

SISTEMA DE GESTIÓN POR MEDIO DE APLICATIVOS MÓVILES PARA EL PROCESO DE ADMINISTRACIÓN, DE SERVICIO AL CLIENTE Y COCINA EN LOS RESTAURANTES

**ANDRÉS MAURICIO VÉLEZ VALENCIA
SANTIAGO GONZÁLEZ MONTOYA**

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, FÍSICA Y CIENCIAS DE LA
COMPUTACIÓN
PROGRAMA DE INGENIERÍA ELECTRÓNICA
PEREIRA
2019**

SISTEMA DE GESTIÓN POR MEDIO DE APLICATIVOS MÓVILES PARA EL PROCESO DE ADMINISTRACIÓN, DE SERVICIO AL CLIENTE Y COCINA EN LOS RESTAURANTES

**ANDRÉS MAURICIO VÉLEZ VALENCIA
SANTIAGO GONZÁLEZ MONTOYA**

**PROYECTO DE GRADO
PARA OPTAR AL TÍTULO DE INGENIERO ELECTRÓNICO EN INGENIERÍA ELECTRÓNICA**

**DIRECTOR
ARLEY BEJARANO, ING. M. SC**

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, FÍSICA Y CIENCIAS DE LA
COMPUTACIÓN
PROGRAMA DE INGENIERÍA ELECTRÓNICA
PEREIRA
2019**

“Principalmente agradezco a Dios, a mis padres por su apoyo incansable en los momentos más ásperos y tenues. A mi madre querida quien es la virtud y voluntad de mi alma.”

Andrés Mauricio Vélez Valencia.

“Los agradecimientos más preciados a quienes me apoyaron en la lucha permanente durante el proceso de formación como ingeniero, a mi familia, amigos y docentes que estuvieron enseñándome el valor de la educación integral.”

Santiago González Montoya.

AGRADECIMIENTOS

En primer lugar, agradecimientos al Ingeniero Arley Bejarano Martínez M. Sc. por la disposición para asesorar el proyecto de la manera más atenta, la colaboración y la ayuda constante.

De igual modo, agradecimientos a la Universidad Tecnológica de Pereira por brindar un camino de oportunidades, de momentos importantes en una gran etapa de la vida, aportando calidad, conocimiento y una visión profesional hacia el mundo. Los recursos académicos brindados por la universidad fueron de vital importancia para el desarrollo del proyecto actual, además, de la excelente educación recibida durante el proceso de formación.

CONTENIDO

AGRADECIMIENTOS	2
LISTA DE FIGURAS	5
RESUMEN.....	6
INTRODUCCIÓN	7
PLANTEAMIENTO DEL PROBLEMA.....	8
JUSTIFICACIÓN	9
OBJETIVOS.....	10
OBJETIVO GENERAL	10
OBJETIVOS ESPECÍFICOS.....	10
MARCO REFERENCIAL	10
ESTADO DEL ARTE	10
MARCO CONCEPTUAL.....	12
APLICATIVO	12
APLICATIVO MÓVIL	12
PÁGINA WEB.....	12
PÁGINA WEB DINÁMICA.....	12
APLICACIÓN NATIVA	12
APLICATIVO WEB O (WEB APP)	12
FRONTEND.....	12
BACKEND	12
SOAP	13
REST.....	13
XML 13	
JSON.....	13
RESTFUL	13
SERVIDOR.....	13
WEB SERVICE.....	13

DISEÑO WEB RESPONSIVO	13
MARCO TEÓRICO	14
BACKEND PARA UNA APLICACIÓN MÓVIL	14
REQUERIMIENTOS PARA DESARROLLAR UN BACKEND	14
BACK-END COMO UN SERVICIO (BAAS)	14
TIPOS DE XAAS	14
VENTAJAS DE UN XAAS.....	15
DESVENTAJAS DE UN XAAS	15
VENTAJAS DE UN BAAS.....	15
DESVENTAJAS DE UN BAAS	15
MBAAS.....	15
FIREBASE.....	16
ANGULAR	16
ANGULARJS.....	16
DESARROLLO.....	17
APLICATIVO MENÚ	18
DISEÑO DEL ALGORITMO DEL APLICATIVO MENÚ.....	22
APLICACIÓN COCINA	27
APLICATIVO WEB DE MONITOREO DE VENTAS, INVENTARIOS Y PEDIDOS.....	31
INTEGRACIÓN DE LOS DOS APLICATIVOS MOVILES Y EL APLICATIVO WEB EN UN ÚNICO SISTEMAS	45
ANÁLISIS Y RESULTADOS	54
PRUEBAS CON DIFERENTES CELULARES	57
CONCLUSIONES	58
TRABAJOS FUTUROS	60
REFERENCIAS.....	61

LISTA DE FIGURAS

Figura 1. Diagrama de flujo del proyecto.	18
Figura 2. Interfaz gráfica de la aplicación menú.	19
Figura 3. Selección de pedidos de la aplicación menú.	20
Figura 4. Selección de mesa junto pedidos de la aplicación menú.	21
Figura 5. Diagrama de flujo de la app menú sección inicio.	22
Figura 6. Diagrama de flujo de la app menú sección service.	23
Figura 7. Diagrama de flujo de la app menú sección cesta de compras.	24
Figura 8. Diagrama de flujo de la app menú sección service.	25
Figura 9. Diagrama de flujo de la app menú sección local storage.	26
Figura 10. Pantalla de los pedidos mostrados en la cocina aplicación cocina.	27
Figura 11. Pantalla de los pedidos mostrados en la cocina aplicación cocina.	28
Figura 12. Pantalla de los pedidos mostrados en la cocina aplicación cocina.	28
Figura 13. Diagrama de flujo de la app cocina sección home.	29
Figura 14. Diagrama de flujo de la app cocina de la sección service.	30
Figura 15. Ventana Home del aplicativo web.	31
Figura 16. Ventana pedida del aplicativo web.	32
Figura 17. Mensaje de alerta pedido.	32
Figura 18. Mensaje de confirmación pedido.	33
Figura 19. Ventana Crear Mesa del aplicativo web.	33
Figura 20. Mensaje de confirmación crear mesa.	34
Figura 21. Ventana Nuevo producto del aplicativo web.	34
Figura 22. Ventana Editar producto del inventario del aplicativo web.	35
Figura 23. Ventana Menú del aplicativo web.	36
Figura 24. Diagrama de flujo de la app web de la sección home.	37
Figura 25. Diagrama de flujo de la app web de la sección creación de mesa.	38
Figura 26. Diagrama de flujo de la app web sección inventario.	39
Figura 27. Diagrama de flujo de la app web sección menú.	40
Figura 28. Diagrama de flujo de la app web sección pedido.	41
Figura 29. Diagrama de flujo de la app web de la sección sistema de cobro.	42
Figura 30. Diagrama de flujo de la app web sección service.	44
Figura 31. Pantallas iniciales de los aplicativos menú, cocina y aplicativo web.	45
Figura 32. Inicio del pedido en el aplicativo móvil de menú.	46
Figura 33. Secuencia del pedido en el aplicativo móvil de menú.	47
Figura 34. Pantallas de los aplicativos menú, cocina y aplicativo web, después del pedido en mesa 2.	48
Figura 35. Secuencia pantallas del aplicativo de cocina para tomar el pedido.	49
Figura 36. Indicación de pedido tomado y cocinando.	50
Figura 37. Secuencia pantallas del aplicativo de cocina para tomar el pedido.	50
Figura 38. Secuencia pantallas del aplicativo de cocina para tomar el pedido.	51
Figura 39. Aplicativo web con pedido listo en la mesa 2.	52
Figura 40. Secuencia pantallas del aplicativo de cocina para tomar el pedido.	53
Figura 41. Secuencia pantallas del aplicativo de cocina para tomar el pedido.	54
Figura 42. Resultados de encuestas para el aplicativo menú.	55
Figura 43. Resultados de encuestas para el aplicativo cocina.	56

RESUMEN

En este documento se presenta el desarrollo de un sistema de gestión para restaurantes por medio de aplicativos móviles para el proceso de administración, de servicio al cliente y cocina en los restaurantes. El sistema se planteó como una posible solución alrededor de las deficiencias de comunicación, administración, accesibilidad e interactividad que tienen los dispositivos y medios tecnológicos usados en los restaurantes.

Las integraciones de aplicativos informáticos enriquecen la estructuración de los procesos de un negocio, para ello se sistematizó una serie de necesidades propias de un restaurante debido a que se evidenciaron problemas organizacionales, administrativos y softwares demasiado complejos para necesidades estándares que hacían perezosa la velocidad, y por lo tanto complejo el dinamismo total de la información. En concreto, el alcance del proyecto intentaba crear una forma más simple y personalizada con la filosofía de divide y vencerás, en otras palabras, hacer uso de pequeños programas interconectados y no de un programa grande y lento que tenga cientos de características, en cambio centrarse en especificaciones de lo más esencial para los restaurantes. Cabe aclarar que se pretendía manejar un cerebro del programa con el aplicativo web y que los aplicativos móviles fuesen los periféricos de este. Así pues, se enfocó el prototipo en flexibilidad para ganar niveles de acceso y sobretodo manejo eficiente del rol de los usuarios. Entonces, se hizo uso de herramientas informáticas para el desarrollo del aplicativo web y los aplicativos móviles; es decir, en el diseño del fronted (lado con el que interactúa el usuario) se utilizaron distintos lenguajes de programación como HTML, CSS y SCSS los cuales se trabajaban en el framework de ANGULAR. Y, en el diseño del backend (lado del servidor) se hizo a través de los servicios web de FIREBASE, además, se utilizaron los lenguajes de programación TYPESCRIPT y JAVASCRIPT, también programables con el framework de ANGULAR.

El trabajo se planeó principalmente en tres etapas, la primera etapa es el diseño aplicativo web que ejecuta un menú digital de pedidos y que se encarga de administrar la base de datos de todas las comidas, bebidas e inventarios de los productos existentes en el stock, con el fin de que un administrador pudiese interactuar con este de forma ágil. En la segunda esta, se diseñó un aplicativo móvil que permite hacer un pedido desde la mesa por el mismo cliente del restaurante para que el pedido llegase al punto central donde trabaja el administrador y luego a la cocina. Todo esto con el fin de que el mesero se centre básicamente en la atención y el asesoramiento para dinamizar los pedidos de manera global en la organización. Y, por último, se diseñó un aplicativo móvil para la cocina del restaurante que enlista los pedidos realizados por el cliente y que cada vez que se termina de preparar un pedido el administrador se enteró inmediatamente de la situación, sin la necesidad de ir hasta la cocina.

El resultado final es increíblemente interesante dado que, a pesar de la ardua programación y codificación en alto volumen, las respuestas del sistema son rápidas. Al mismo tiempo, existe simplicidad de entendimiento para cualquier persona, igualmente, el sistema es multiplataforma y responsivo lo que es una ventaja enorme a la hora de la implementación. Se destaca que puede incrementar su escalabilidad rápidamente, la base de datos otorga reacciones instantáneas cuando el sistema está procesando grandes cantidades de pedidos sin problema alguno. Y, lo más importante es que en el futuro se pueden agregar mejoras en las versiones como por ejemplo la adición de notificaciones push, el manejo de timers para los pedidos, chat en línea para los clientes (servicio al consumidor), método de pago desde la mesa

y generación de informes de tendencias de consumo con analíticas para mejorar la rentabilidad del restaurante. En conclusión, el uso de la programación en la web y la programación de móviles nativos e híbridos está en un direccionamiento avasallador, enlazado con la electrónica, el internet de las cosas está cambiando nuestro mundo a un nivel tal que se podrá automatizar fácilmente los servicios de restaurantes también con robots, redes de comunicaciones y dispositivos programables.

INTRODUCCIÓN

Para los negocios actuales la información es notablemente valiosa siendo un factor diferenciador de los negocios exitosos. En una organización se puede manejar la información de muchas maneras, por ejemplo, según los servicios o por los productos ofrecidos a los clientes. Una herramienta que permite gestionar tareas relacionadas con los productos y servicios comercialmente, tanto físico como online se llama un TPV (terminal punto de venta), el cual ayuda a gestionar las compras, clientes, facturas y organiza las labores comerciales. El software TPV permite diversas funciones como contabilizar pedidos y productos, realizar cobros, control de stock, administración de productos, movimiento de inventarios, verificación de caja, entre otras.

El software TPV articula las operaciones en tiempo real con la capacidad de crear bases de datos de clientes y proveedores, manejando el estado de los productos, también emitiendo estadísticas y generando informes y facturación. El TPV agiliza las ventas para brindar un buen servicio a los clientes, en consecuencia, sin errores de contabilidad o listas incompletas en el stock. El software permitirá optimizar los movimientos en la caja ahorrando tiempo y dinero.

El software TPV es una piedra angular que construye un camino rentable para los restaurantes. Su empleo no necesita conocimientos informáticos. De modo que, este sistema robusto atenderá al cliente de una forma más rápida y personalizada.

El desarrollo de un software TPV es una alternativa interesante si se piensa de manera en que se puedan distribuir las funciones requeridas por un cliente en especial. Por eso para los restaurantes estos medios tecnológicos se están requiriendo cada vez más, dado su amplia gama de utilidades y ventajas que generan un excelente valor dentro de la evolución de este tipo de negocio.

PLANTEAMIENTO DEL PROBLEMA

La industria gastronómica en Colombia está en crecimiento actualmente, después de un estancamiento de dos años que ha tenido el sector. Una de las problemáticas que tiene el sector gastronómico en el país es la tecnología, puesto que, se ha visto un rezago tecnológico y los establecimientos colombianos no han querido optar por mejorar la calidad de su servicio por medio de nuevas tendencias tecnológicas que pueden dinamizar su negocio de una manera mucho más eficiente y eficaz [1].

Aunque en Colombia, los dueños de restaurantes, bares, café shops, dueños de establecimientos de comida rápida y demás, están utilizando nuevas tendencias como los es rappi y uber eats como medidas para una nueva forma de entrega de pedidos a domicilio, las demás tecnologías como por ejemplo los TPV's (Terminal punto de venta) digitales son muy bien utilizadas en el sector [1] [2].

El proyecto de gestión e integración de aplicativos web y móviles viene a ofrecer una alternativa de solución a la permanente necesidad del sector de los restaurantes de manejar grandes cantidades de información en cuanto a clientes, servicio, innovación, publicidad y administración de recursos. Por lo tanto, es una premisa tener las bases de datos actualizadas acerca de informes, cuentas, contabilidad, productos y sistema de consumo a través de analíticas, lo que hace posicionar el negocio fuertemente en el mercado de las apps tipo TPV y un posicionamiento del mercado local y global de los restaurantes.

Una dificultad constante radica en el manejo de gran cantidad de información que genera el funcionamiento completo de los restaurantes, lo que entorpece la posibilidad de brindar una organización rápida, eficiente y oportuna. Además, el tiempo de elaboración de los distintos informes aumenta haciendo que el proceso sea dispendioso.

Actualmente los sistemas de gestión para los restaurantes carecen de buenos sistemas de integrados de gestión debido al bajo soporte de las empresas creadoras del software, y algunas de sus aplicaciones más destacadas son de pago o estilo freemium, pero su mayor desventaja es la complejidad del entendimiento de los sistemas y su acople a las características de cada negocio, se demanda mucho tiempo de adecuación a las necesidades empresariales. Hiboutik, Kikli, TPV Gratuito y VisionWin son empresas que han incursionado en el área de TPV, [3].

Estas empresas ofrecen servicios de contabilidad, comercio electrónico, servicio de pagos, reservas, marketing, administración y demás, para cualquier tipo de negocios PYMES y MY-PIMES. Cada una de estas empresas han sido creadas y desarrolladas en España, a diferencia de Hiboutik, la cual fue creada por una casa de modas francesa. Por otro lado, Camarero 10 es un TPV creado exclusivamente para brindar servicios de gestión de restaurante y bar. Este es desarrollado por Digital Restaurant, empresa mercantil española con domicilio en Madrid España.

Por todo esto, se plantea ¿Cuáles son las deficiencias de comunicación, administración, accesibilidad e interactividad que tienen los dispositivos y medios tecnológicos usados en el interior de los restaurantes colombianos?

JUSTIFICACIÓN

En los restaurantes hay muchas formas en las que un negocio puede volverse más robusto con ideas tecnológicas de ingeniería moderna; si se quiere una mayor velocidad de procesamiento de información y tener espacios para una administración vertical de las complejas dinámicas que se tienen en los restaurantes, el presente proyecto se perfila como una solución potencial.

Hoy en día existe una gran variedad de herramientas tecnológicas que ofrecen diferentes soluciones a una gran cantidad de problemáticas presentes en las sociedades contemporáneas, así como también, a las necesidades diarias de las personas. Una de las herramientas tecnológicas más usadas actualmente por las empresas, industrias y personas en general, son los celulares inteligentes o smartphones. Estos dispositivos, además de ser útiles, portables y fáciles de usar, brindan una gran cantidad de funciones que permiten realizar tareas diarias de las personas de forma fácil e interactiva, tales como, realizar pagos desde cualquier parte del mundo, tomar fotografías de forma fácil, comunicación con otras personas en cualquier parte del mundo de manera ágil, entre otras, [4].

Además de los celulares, otro de los dispositivos tecnológicos más utilizados por las empresas, industrias y personas en general, son los computadores portátiles y de escritorio. Estos ofrecen una gran cantidad de funciones que permiten a las empresas, industrias y negocios en general realizar tareas indispensables de manera ágil y segura. Por esto, emplear smartphones y computadores es una buena forma para mejorar la comunicación e interacción entre las empresas y sus clientes o potenciales clientes futuros [4] [5].

Los sistemas de información a través de aplicaciones nativas, web apps y web apps nativas, desempeñan alternativas versátiles y funcionales en procesos de desarrollo de gestión de unidades de negocio con criterios de calidad. El uso inteligente de un *backend* y las posibilidades de utilizar un *frontend* llamativo, interactivo se direccionan como una ayuda a la carga recurrente de escalabilidad. Para lograr una fiabilidad en los sistemas de gestión integrados se requieren de procesos sistemáticos que puedan agilizar la entrega oportuna de servicios de alto nivel a los usuarios. Es importante pensar en la simplicidad y la estructuración cuando se debe estandarizar una necesidad empresarial porque cada tipo de negocio tiene un nivel de competitividad y complejidad propia.

Es por esto que se diseñó un sistema de gestión por medio de aplicativos móviles y aplicativo web para el proceso de administración, de servicio al cliente y cocina en los restaurantes. Por medio de estos sistemas se busca mejorar la eficiencia y comunicación de los procesos internos de administración, servicio al cliente y cocina de los restaurantes. Ya que, para que haya un correcto funcionamiento de toda la estructura que compone a un restaurante, cada una de las áreas del restaurante debe trabajar de manera sincronizada y bien comunicada, para poder brindar un buen servicio al cliente y una excelente calidad.

Por medio del proyecto descrito anteriormente se logró dar solución a los problemas de comunicación entre las áreas de cocina, administración y servicio al cliente. Por medio de dos aplicativos móviles y un

aplicativo web interconectados entre sí, se logró dar una comunicación rápida y clara del estado de cada una de las áreas del restaurante. Además, le brinda al área de administración del restaurante una herramienta fácil de usar que agiliza las tareas de cobro, revisión y administración de inventarios y además permite coordinar el restaurante en general.

OBJETIVOS

OBJETIVO GENERAL

Desarrollar un sistema de gestión por medio de dos aplicativos móviles para el proceso de administración, de servicio al cliente y cocina en los restaurantes.

OBJETIVOS ESPECÍFICOS

- Desarrollar un aplicativo móvil para que el cliente realice pedidos de comidas y bebidas de los productos ofrecidos por un restaurante.
- Programar un aplicativo móvil de recepción y confirmación de pedidos para la cocina de un restaurante.
- Desarrollar un aplicativo web que permita monitorear las ventas, inventarios y pedidos.
- Integrar los dos aplicativos móviles y el aplicativo web en un único sistema.

MARCO REFERENCIAL

ESTADO DEL ARTE

En los años 80 se presentaban problemas en la comunicación de información entre diferentes ordenadores, por lo que compartir información era una tarea extenuante y lenta. En 1989, Tim Berners Lee, un empleado de CERN (Organización Europea para la investigación Nuclear), tomando en cuenta los problemas que tenían los físicos de la organización para compartir información, escribía una propuesta para una base de datos de hipertexto con enlaces tipados, pero esto genero poco interés [6].

En los años 90, en la necesidad de crear estándares para la comunicación entre varias máquinas, se realizaron varios intentos, entre estos CORBA (definido por Object Management Group OMG) y DCOM (Creado por Microsoft), pero todos estos fracasaron. La idea de los desarrolladores era la creación de aplicaciones web que se puedan escalar y crecer añadiendo más máquinas, pero estos estándares presentan complicaciones con la implementación de tecnologías de las maquinas que se pretenden comunicar y las reglas del firewall, las cuales cortan el tráfico de datos. Por otro lado, en 1990 Tim Berners ya había desarrollado las herramientas protocolo de transferencia de hipertexto HTTP, HTML, el primer navegador web (WorldWideWeb) y el primer servidor de aplicaciones HTTP y las primeras páginas web que describirían el mismo proyecto. En 1991 Berners publicó un breve resumen de World Wide Web, lo que marcaría el inicio de la web como un servicio público [7].

En 1993 un grupo de 8 desarrolladores comenzaron a trabajar en la mejora del dominio NCSA HTTP y

en el año 1995 fue lanzado APACHE en un esfuerzo para desarrollar y mantener un servidor HTTP de código abierto, cofundado por 21 personas, entre estas Brian Behlendorf (Tecnólogo, ejecutivo y programador Americano, fue el primer desarrollador de APACHE ser- ver) 4 y Roy T. Fielding (del cual se hablara más adelante), en abril de 1996 APACHE llego a ser uno de los servidores web más importantes en la internet y el 1 de junio de 1999 fue efectiva la incorporación de la APACHE software Foundation, [8].

En 1996 las compañías entendieron que el uso de la Web no sería una opción más, la familiarización de la gente con la Web condujo a las compañías a ver una oportunidad clara en el comercio electrónico y cada vez, más compañías fueron añadidas dentro de la Web. En la necesidad de crear una comunicación más efectiva entre maquinas, en el año 1999 se comenzó a crear nuevos estándares de comunicación. Las grandes compañías, tales como IBM, Microsoft, Verisign y demás, comenzaron con sus propios desarrollos, resultando en una cantidad enorme de siglas y tecnologías complejas (XML, SOAP, WSDL, UDDI y muchas más), cuyas implementaciones fueron incompatibles durante años [6].

En el año 2000, Roy T. Fielding, en su tesis doctoral. Estilos Arquitecturales y el Diseño de Arquitecturas de Software basadas en Red describe Transferencia de Estado Representacional (REST) como una alternativa diferente a las ya utilizadas como CORBA o DCOM, dando un nuevo enfoque a los servicios web, creándose de esta manera las Web API's como un conjunto coherente de operaciones mediante HTTP. Unos de los diseños fundamentales de REST son los recursos de información, en los que HTTP como un conjunto de operaciones POST, PUT, GET y DELETE [9].

En 1999, Marc Benioff, un programador y ex vicepresidente de ORACLE funda la compañía Salesforce, una empresa de software bajo demanda la cual produce CRM (customer relations- hip management) por suscripción, en el año donde la web services apenas comenzaba a florecer. La característica principal de los productos de Salesforce fue la facilidad para enlazarse con cualquier otro sistema de la empresa. Fue técnicamente la primera compañía en vender una API como parte de sus servicios en el año 2006, pero pocos equipos de desarrollo pudieron aprovechar su complicada API XML. Por otro lado, Ebay a los finales del año 2000 implementó una REST API accesible y fácil de implementar, su intención era impulsar el área de las soluciones de comercio electrónico, pero de paso fomentaron el área de las API's [10]. Con el crecimiento de la internet y todas las oportunidades que las API's aportaban para el desarrollo de nuevas plataformas empresariales, en agosto de 2004, Flickr creo su propia API REST, en el momento indicado para el auge de las redes sociales y los blogs. Rápidamente la API de Flickr se convirtió en la plataforma favorita de los bloggers para el manejo de imágenes en sus sitios.

En noviembre de 2004, Amazon lanzo su servicio web service llamado Simple Queue Service, el cual sirve para enviar grandes cantidades de datos y archivos, además de dar soporte a empresas que trabajan con grandes cantidades de datos, tales como Netflix, Dropbox y Nextdoor. Pero Amazon Web Services (AWM) nació como tal en 2006, aportando a los desarrolladores una tonelada de espacio de datos. En 2007 Amazon Web Services ya tenía más de 300.000 desarrolladores inscritos, extendiéndose en noviembre del mismo año a la Unión Europea y en 2011 a Sao Paulo y Brasil, ofreciendo así sus servicios a toda Latinoamérica.

En 2014 Google adquiere Firebase, una plataforma de desarrollo de aplicaciones web desarrollada por James Tamplin y Andrew Lee en 2012. En 2015, Google adquiere Divshot, una plataforma para crear interfaces gráficas de aplicaciones web de forma fácil y cómoda para desarrolladores, integrando las funcionalidades de Divshot a los servicios de Firebase [11]. Desde el 2006 las API's han cambiado la web de una forma drástica, mejorando el entorno para los desarrolladores, creando nuevas oportunidades de negocio [12]. Un sector económico en el que se han estado posicionando las API's, son la creación de aplicativos web para PYMES y MIPYMES de cualquier tipo de negocios. En 1988, IGT Microelectronics, incorporó ordenadores para agilizar el proceso mercantil en las tiendas, decisión que comenzó a revolucionar el comercio en la década de los 90's. De este modo, IGT incursiono en el primer TPV

(terminal punto de venta). En 1993 TPV Axel tuvo gran aceptación en farmacias, y desde esta fecha, los TPV han evolucionado constantemente, incurriendo en la utilización de API's. Mediante API's, los TPV's tuvieron un gran desarrollo, extendiendo su participación en el mercado no solo a cajas registradoras, sino también, a softwares que mejoran la gestión mercantil de cualquier negocio [13]. En 2012, fue creado Hiboutik, el primer software TPV gratis a nivel mundial. Hi- boutik está integrado con diferentes aplicaciones de contabilidad, comercio electrónico, pagos y marketing. El desarrollo de su API le permite conectarse con las funciones de Paypal, Lydia, Google, shopify y otras más [3]. Así como Hiboutik, diferentes empresas han creado sus propias API's y aplicativos web para brindar sus servicios a diferentes negocios, y en particular, servicios de hostelería para restaurantes. Klikin, VisionWin y TPV Gratuito 123, por ejemplo, son empresas españolas de desarrollo web y app que ofrecen servicios de pagos, pedidos, reservas, promociones, marketing y administración para todo tipo de PYMES. Otro ejemplo de TPV es Camarero 10, el cual ofrece servicios únicamente para hostelería y restaurante. Provee a sus clientes servicios de administración, pagos, pedidos, creación de informes, gestión de sedes y demás.

MARCO CONCEPTUAL

APLICATIVO

Programa diseñado para ejecutar una o múltiples tareas a través de un dispositivo electrónico.

APLICATIVO MÓVIL

Programa diseñado para ejecutar una o múltiples tareas a través de un dispositivo electrónico móvil.

PÁGINA WEB

Página digital multimediatca de estándar (world wide web), accesible a través de un navegador web y esta almacenada en un servidor.

PÁGINA WEB DINÁMICA

Es una página web que permite la construcción de contenido dinámico al momento de hacer una petición usando recursos como bases de datos que requieren un aplicativo web.

APLICACIÓN NATIVA

Programa diseñado para ejecutar una o múltiples tareas adaptables a una plataforma específica y requiere ser instalada en un dispositivo electrónico.

APLICATIVO WEB O (WEB APP)

Programa diseñado para ejecutar una o múltiples tareas que se carga en un servidor web y se interpretan en un navegador de cualquier sistema operativo sin necesidad de instalarse en un dispositivo electrónico.

FRONTEND

Sección de un aplicativo que se encarga directamente del acceso y la interactividad con los usuarios.

BACKEND

Comprende la parte lógica de un aplicativo que se conecta una base de datos y el servidor, y no es directamente accesible por los usuarios.

SOAP

(Simple Object Access Protocol) protocolo para la comunicación entre servicios con contratos (WSDL: Web Services Description Language), y se basa en XML.

REST

(Representational State Transfer) es un protocolo para la comunicación entre servicios con contratos (WADL: Web Application Description Language), basada en XML JSON sobre HTTP.
HTTP

(HyperText Transfer Protocol) protocolo de transmisión de información de la world wide web para "hablar" un mismo idioma a la hora de transmitir información por la red.

XML

(Extensible Markup Language) metalenguaje estándar que permite la organización y el etiquetado de documentos y estructura el intercambio de información entre diversas plataformas.

JSON

(JavaScript Object Notation) formato ligero de intercambio de datos basado en el lenguaje de programación de JavaScript.

RESTFUL

Uso de un WEB SERVICE por medio del protocolo REST aplicando las operaciones estándar GET, POST, PUT, DELETE, entre otras.

SERVIDOR

Ordenador que se encarga de gestionar información de diferente índole a diversos clientes a través de una arquitectura cliente-servidor.

WEB SERVICE

Son múltiples tipos de servicios web que permiten intercambio de mensajes y comunicación de aplicativos de diferentes características por medio de protocolos SOAP y REST.

DISEÑO WEB RESPONSIVO

Estándar para la adaptación de la apariencia de páginas web a dispositivos electrónicos tales como tabletas, teléfonos inteligentes, computadores, laptops, entre otros.

MARCO TEÓRICO

Una década atrás, para comercializar un producto o servicio globalmente era necesario tener un sitio web. Actualmente un sitio web no es suficiente también es necesaria una aplicación móvil. Para comercializar un producto con éxito es importante hacer uso de una aplicación móvil porque contiene características de procesamiento más simples para el cliente. Para aplicaciones móviles y web se habla de frontend y backend. El frontend define la interfaz de usuario, la interacción del usuario y la presentación de la información. El frontend es el navegador web o dispositivo móvil de los usuarios, mientras que el backend es un servidor o los servidores en donde se almacenan y se comparten los datos [14].

BACKEND PARA UNA APLICACIÓN MÓVIL

Un backend para una aplicación móvil es un elemento que administra la seguridad, el almacenamiento de datos y la lógica empresarial. Por ejemplo, un backend permite:

- Uso de analíticas.
- Seguridad extendida.
- Manejo de publicidad y anuncios.
- Comunicación entre usuarios (juegos de multijugador o aplicaciones de mensajería).
- Notificaciones push.
- Monetización de la aplicación a través de anuncios

REQUERIMIENTOS PARA DESARROLLAR UN BACKEND

Para integrar un aplicativo móvil a un backend es necesario conocer la infraestructura de red, servidores web, software de gestión de bases de datos, lenguajes de scripts de servidor, criptografía moderna, seguridad informática, big data, minería de datos, redes de telecomunicaciones móviles y tecnologías adicionales [6].

BACK-END COMO UN SERVICIO (BAAS)

Xaas es una palabra que indica una solución empresarial total como servicio o producto TIC y las empresas pueden rentar su uso [15-16].

TIPOS DE XAAS

- SaaS (Software como Servicio)
- PaaS (Plataforma como Servicio)
- IaaS (Infraestructura como Servicio)
- Hardware como Servicio (HaaS, hardware as a service)
- Almacenamiento como Servicio (SaaS, storage as a service)
- Base de Datos como Servicio (DBaaS, database as a service)

- Recuperación ante Desastres como Servicio (DRaaS, disaster recovery as a service)
- Comunicaciones como Servicio (CaaS, communications as a service)
- Red de trabajo como Servicio (NaaS, network as a service)

VENTAJAS DE UN XAAS

- Inversión inicial baja.
- Sin gastos de compra.
- Sin costes de implementación.

DESVENTAJAS DE UN XAAS

- Coste de suscripción.
- Costo fijo por usuario o por consumo.
- Falta de confiabilidad, debido que se está dependiente del proveedor XaaS.
- Sin garantía de la continuidad de una suscripción.

XaaS está redefiniendo la forma en que se desarrolla, publica, y se consume un software. Es decir, es como tener las posibilidades de un desarrollo de backend como un servicio web. Los proveedores de BaaS ofrecen varias características a través de modelos de precios, como por ejemplo un modelo "freemium" que permite el almacenamiento de datos, uso de analíticas y usuarios, autenticación y notificaciones push las cuales se proporcionan de manera gratuita con cierto límite. Pero, cuando se solicitan funciones adicionales, entonces, se cobra una tarifa ayudando a la construcción de una aplicación gratis, para luego escalarla a un nivel de pago a medida que se adicionan clientes. Usualmente, el desarrollador requiere los SDK y las API del proveedor BaaS para enlazar su aplicación al backend [17-18].

VENTAJAS DE UN BAAS

- No es necesario construir y administrar backends.
- Evitar aprendizaje asociado a las tecnologías de backend. Reducir el costo y el tiempo de desarrollo.
- Bajo costo para la experimentación de aplicaciones nuevas en el mundo real.

DESVENTAJAS DE UN BAAS

- El proveedor de BaaS puede cerrar el servicio.
- Si se requiere migrar a otro proveedor BaaS, quizás, será necesario rediseñar la aplicación porque puede tener una API diferente.
- Limitación de la personalización de la infraestructura del backend en un BaaS, es decir, muchas funciones deseadas para una aplicación no están disponibles [19].

MBAAS

La computación de servicio en la nube puede consumir un servicio mientras se administra un backend móvil permitiendo:

- Asegurar infraestructura y almacenamiento.
- API, incluidas las redes sociales y varias más, según los requisitos de la aplicación.
- Capacidad de navegar sin problemas en varias plataformas y dispositivos móviles.
- Autenticación y gestión de usuarios.
- Almacenamiento de persistencia de datos.
- Notificación push.
- Servicios de geolocalización.
- Analíticas.
- Aprovechamiento de la arquitectura de microservicios.
- Facilidad de uso y excelente soporte al cliente.

FIREBASE

La plataforma BaaS de Google, ofrece funciones las cuales se dividen en tres etapas: Desarrollar, Ganar y Crecer. La etapa de desarrollo consiste en bases de datos en tiempo real, autenticación, mensajería en la nube, almacenamiento, alojamiento, laboratorio de pruebas (para probar aplicaciones en un dispositivo) e informes de fallos. Además, la plataforma AdMob tiene la función de monetizar la aplicación. Firebase es también una plataforma Android. Y, el SDK de Firebase está para iOS [20].

ANGULAR

Angular es un framework para el desarrollo de aplicaciones web, el cual está diseñado en Javascript y es de código abierto [21].

ANGULARJS

La lógica de presentación de la información se ha movido del servidor hacia los clientes, el servidor se ha optimizado de trabajo, dado que solo debe enviar los datos a través de JSON al cliente y es éste a su vez construirá el HTML que sea necesario, Además, es más ligero transferir datos simples que el HTML completo. Un frameworks como AngularJS crea aplicaciones rápidas por disponer de un conjunto de librerías, con un conjunto paradigmas y patrones que facilitan el desarrollo del software y sobre todo su mantenimiento [22].

El llamado MVC, es decir, la separación del código en diferentes responsabilidades situando cada parte del código en un sitio determinado, y ese orden facilita que los desarrollos sean más manejables siendo más potente, sencillo y extensible [23-24].

AngularJS tiene amplias opciones para hacer aplicaciones web, aplicaciones de negocios, aplicaciones de gestión de información, e incluso desarrollo de apps similares a las nativas, también como aplicaciones de escritorio de frontal web [25-33].

DESARROLLO

Para el planteamiento de todo el proyecto se tuvo en cuenta las necesidades de comunicación que tienen las áreas de administración, cocina y servicio al cliente en un restaurante. Una vez sabido esto, el proyecto se dividió en 4 etapas, diseño del aplicativo web, diseño del aplicativo menú, diseño del aplicativo de cocina y por último la interconexión de cada aplicativo en un solo sistema.

El funcionamiento del sistema completo funciona por eventos, es decir, cada vez que el sistema detecta un evento o un cambio hecho por el usuario, el sistema realiza las tareas correspondientes. Cada vez que se detecta un evento en cualquiera de los aplicativos, estos llaman al servidor de firebase. Por medio del servidor, cada uno de los aplicativos saben el estado completo del sistema, de esta forma, cada aplicativo se interconecta y realiza las tareas correspondientes a cada evento realizado por el usuario. Por ejemplo, si el administrador ingresa un nuevo producto en el aplicativo web, este producto se verá en el aplicativo de menú, si un cliente pide este producto en un pedido, este producto se verá en el aplicativo de la cocina.

Como se puede observar en la figura 1, cada vez que uno de los aplicativos llaman al servidor de firebase se envía información, y a la vez, el servidor retornara cierta información. El tipo de información que se envíe entre el aplicativo y el servidor dependerá del evento realizado por el usuario.

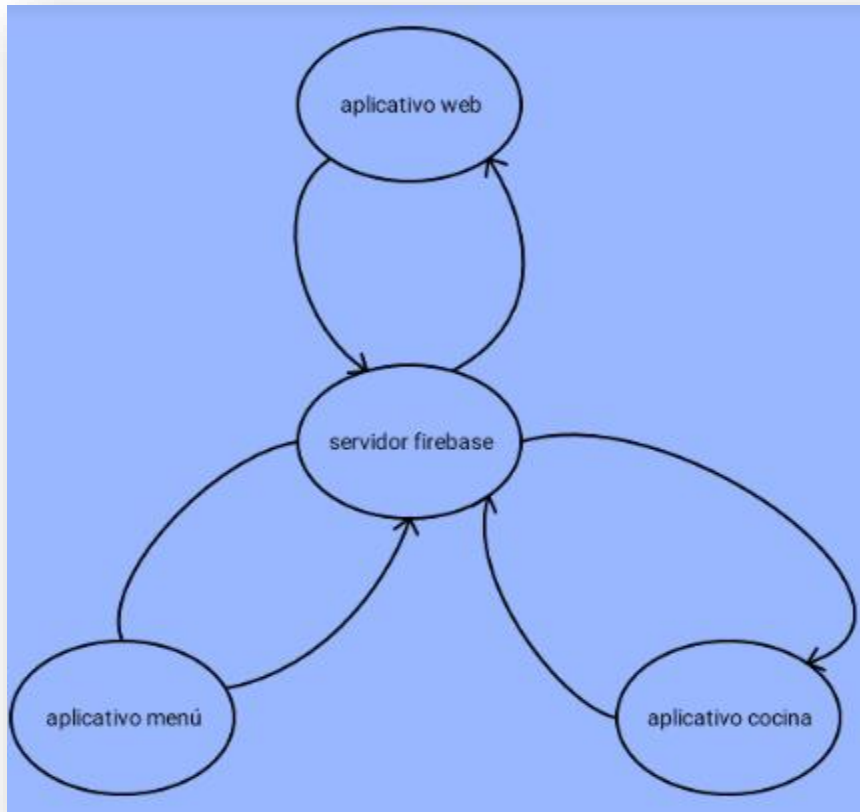


Figura 1. Diagrama de flujo del proyecto.

APLICATIVO MENÚ

En el diseño de la aplicación móvil de pedidos para un restaurante se establecieron en primera instancia las necesidades enlistadas a continuación:

- El cliente visualiza el menú.
- El cliente selecciona la mesa en donde realiza el pedido.
- El cliente selecciona que producto desea, entonces va a los detalles del producto y puede agregar al carrito (pedido) o regresa al menú principal.

Si selecciona el producto, es decir, agrega el producto al carrito (pedido) entonces queda guardado en la base de datos local del dispositivo móvil. Por lo tanto, se dirige al menú principal.

El cliente puede seleccionar otro producto o dirigirse al panel de carrito (pedido) de compras.

Si el cliente decide ir al carrito de compras, se abrirá una ventana en donde visualizará los productos que ha agregado anteriormente al carrito de compras.

En la ventana de carrito de compras, el cliente puede seleccionar los productos que desea pedir, así como también la cantidad que desea, y, puede visualizar el valor total del pedido.

Al hacer clic en comprar se le pide confirmar si desea realizar el pedido, en caso de confirmar el pedido se enviará a la base de datos de firebase, y será dirigido al menú principal, como se muestra en las figuras 1,2 y 3.



Figura 2. Interfaz gráfica de la aplicación menú.

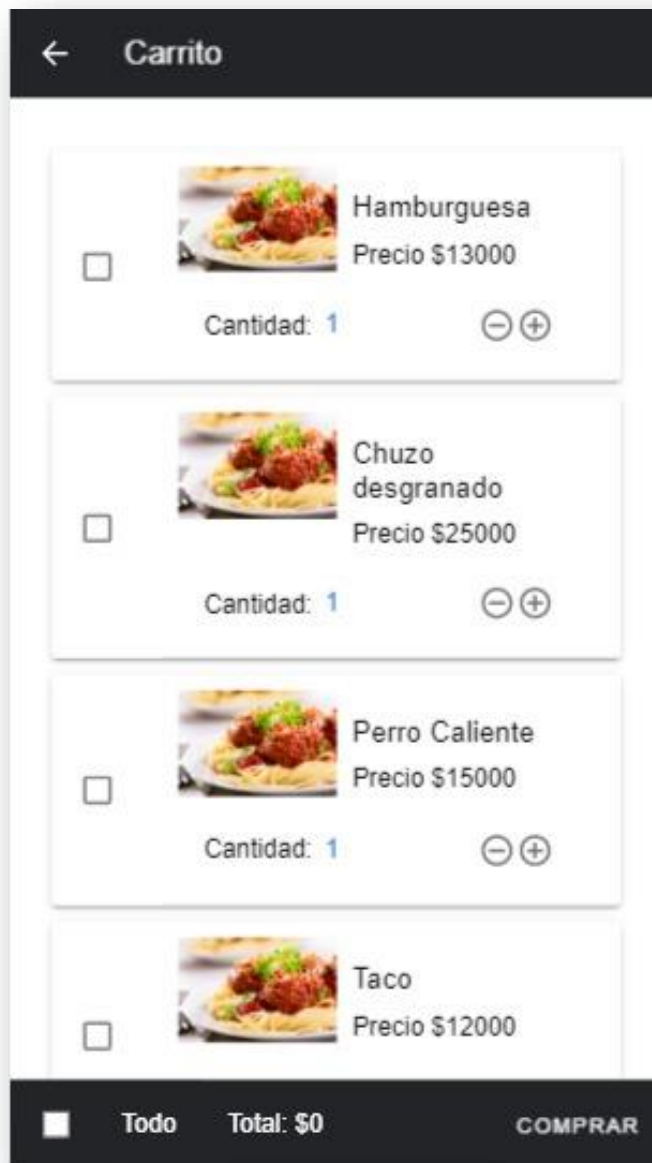


Figura 3. Selección de pedidos de la aplicación menú.



Figura 4. Selección de mesa junto pedidos de la aplicación menú.

DISEÑO DEL ALGORITMO DEL APLICATIVO MENÚ

El diagrama de flujo del sistema de inicio trabaja de la siguiente manera:

Se debe iniciar y entrar a la página principal donde se muestra una lista, los menús. Si se da clic en alguno de los menús, entonces se dirige a la pantalla de detalles. En el encabezado está el icono de un carrito de compras y al dar clic en este sitio, entonces se dirige a la pantalla de carrito donde están todos los pedidos preseleccionados. En la figura 4 se muestra el diagrama de flujo explicativo.

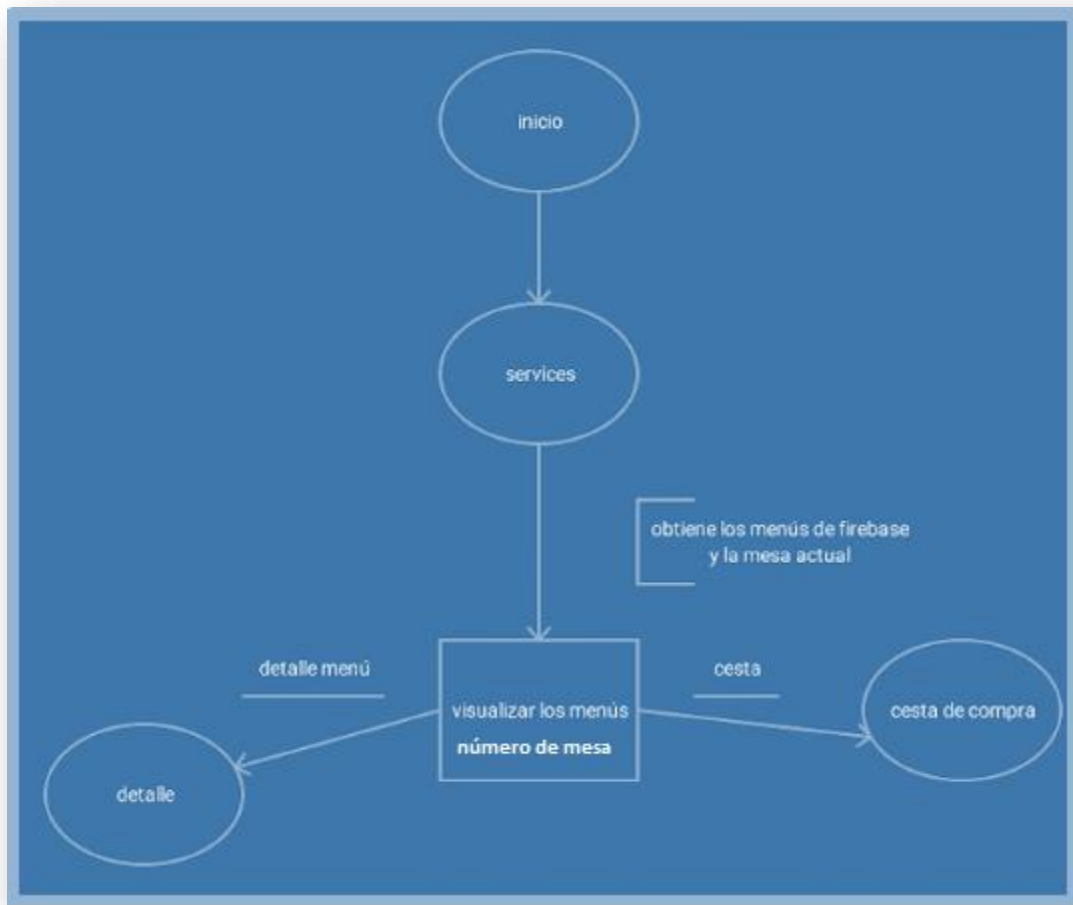


Figura 5. Diagrama de flujo de la app menú sección inicio.

El diagrama de flujo del sistema de detalle trabaja de la siguiente manera:

Al entrar a detalles se hace el llamado a la información de firebase del producto seleccionado y se muestra toda la información del producto. Luego, se muestra toda la información del producto como la imagen principal, el nombre, precio, descripción general y la galería de imágenes. Sí se le da al botón de regresar, entonces va a la pantalla principal, y, al dar clic a añadir a la cesta, luego, la confirmación mostrada es si desea añadir el producto al carrito, por lo tanto, y, si confirma el proceso, se añade al carrito, posteriormente, regresando a la pantalla principal. En la figura 5 se muestra el diagrama de flujo explicativo.

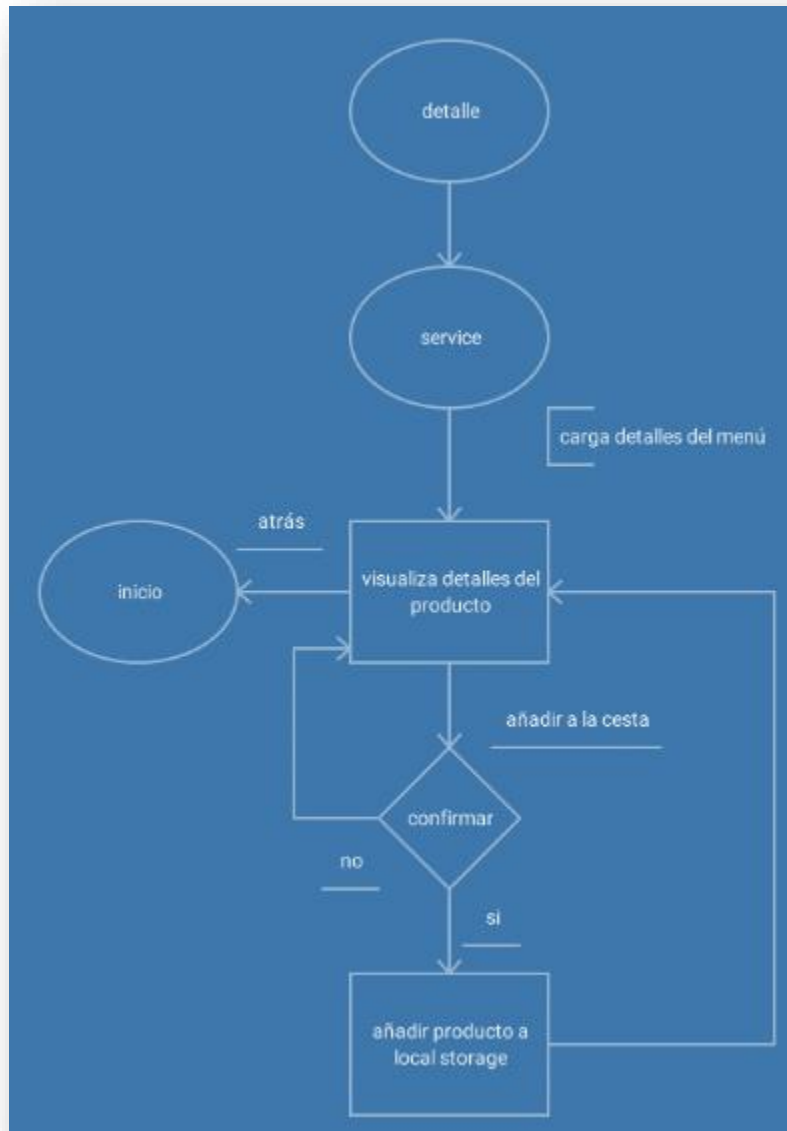


Figura 6. Diagrama de flujo de la app menú sección service.

El diagrama de flujo del sistema de cesta de cobro trabaja de la siguiente manera:

En la pantalla del carrito de compras, se muestra todos los productos preseleccionados a pedir. El cliente puede chequear o seleccionar uno o varios productos y puede eliminarlo, aumentar o disminuir la cantidad que desea pedir, ver los detalles del producto y puede realizar el pedido. Si selecciona el icono de basurero, se le pregunta si desea eliminar todos los productos seleccionados del carrito de compras. En la parte central del pie de página, se muestra el subtotal del costo del pedido. La parte inferior derecha, se encuentra el botón de realizar pedido o compra. Si da clic en el botón de comprar, se le pregunta al cliente si desea realizar el pedido, y si da aceptar, se modifica la base de datos de firebase con el pedido que el cliente realizo. En la figura 6 se muestra el diagrama de flujo explicativo.

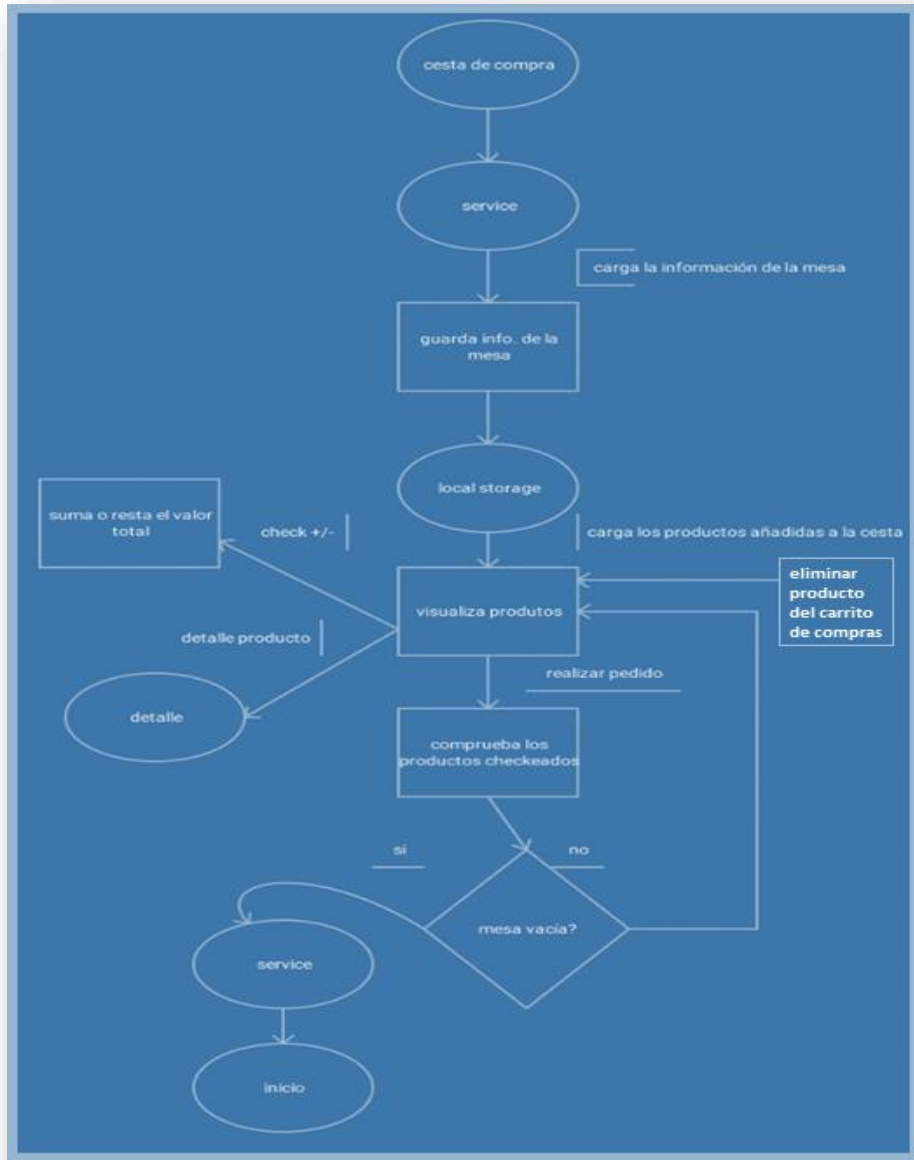


Figura 7. Diagrama de flujo de la app menú sección cesta de compras.

El diagrama de flujo del sistema de service trabaja de la siguiente manera:

Al entrar en la sección de carrito de compras, firebase envía la información de la mesa seleccionada en la sección de menú principal. Luego carga los productos que el cliente envió a carrito de compras anteriormente y los visualiza. Al seleccionar - o + de los productos seleccionados, aumenta o disminuye el valor del costo total, Si selecciona la información del producto, se envía a la sección de detalles. Si se selecciona eliminar, elimina los productos seleccionados. Si selecciona comprar, primeramente, comprueba si la mesa está vacía, si no está vacía, se devuelve a mostrar los productos de carrito de compras. Si la mesa está vacía, envía la información del pedido al servidor firebase y luego se devuelve a la pantalla principal. En la sección de service, se encuentran 4 funciones, add pedido, get pedido, menú y pedidos, En la figura 7 se muestra el diagrama de flujo explicativo.

- **Add pedido:** añade un nuevo documento a la colección de pedidos.
- **Get pedido:** obtiene un nuevo documento a la colección de pedidos.
- **Menú:** obtiene todos los documentos de la colección de menús.
- **Pedidos:** obtiene todos los documentos de la colección de pedidos

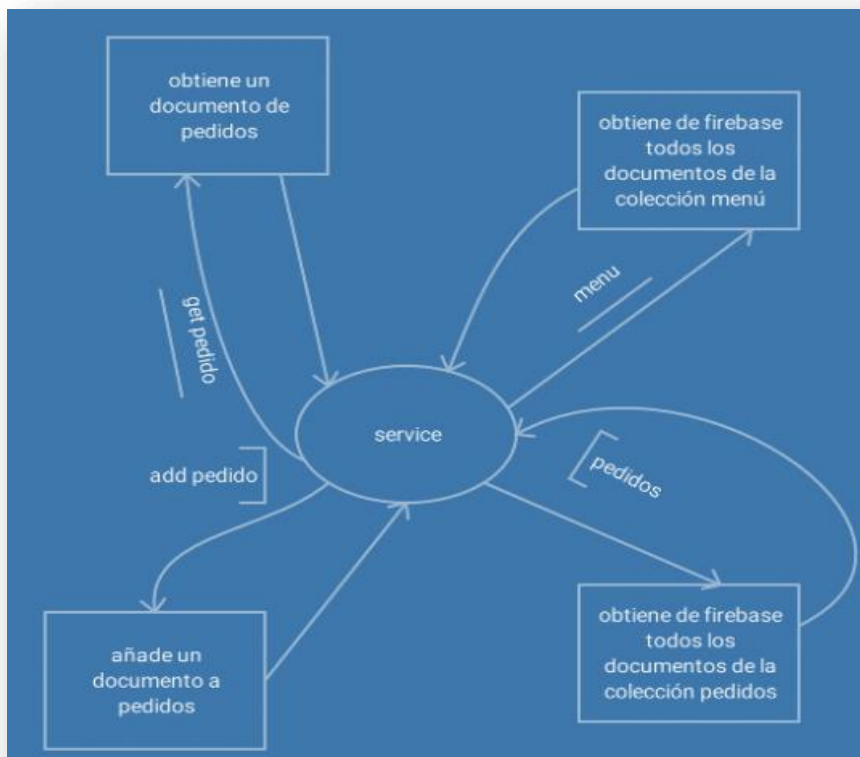


Figura 8. Diagrama de flujo de la app menú sección service.

El diagrama de flujo del sistema de local store trabaja de la siguiente manera:

En la sección del local Store hay 2 funciones, cargar menús y guardar menús. La función cargar menús obtiene la información de los productos añadidos al carrito de compras. La función guardar menú guarda un nuevo producto añadido al carrito de compras en el local store, En la figura 8 se muestra el diagrama de flujo explicativo.

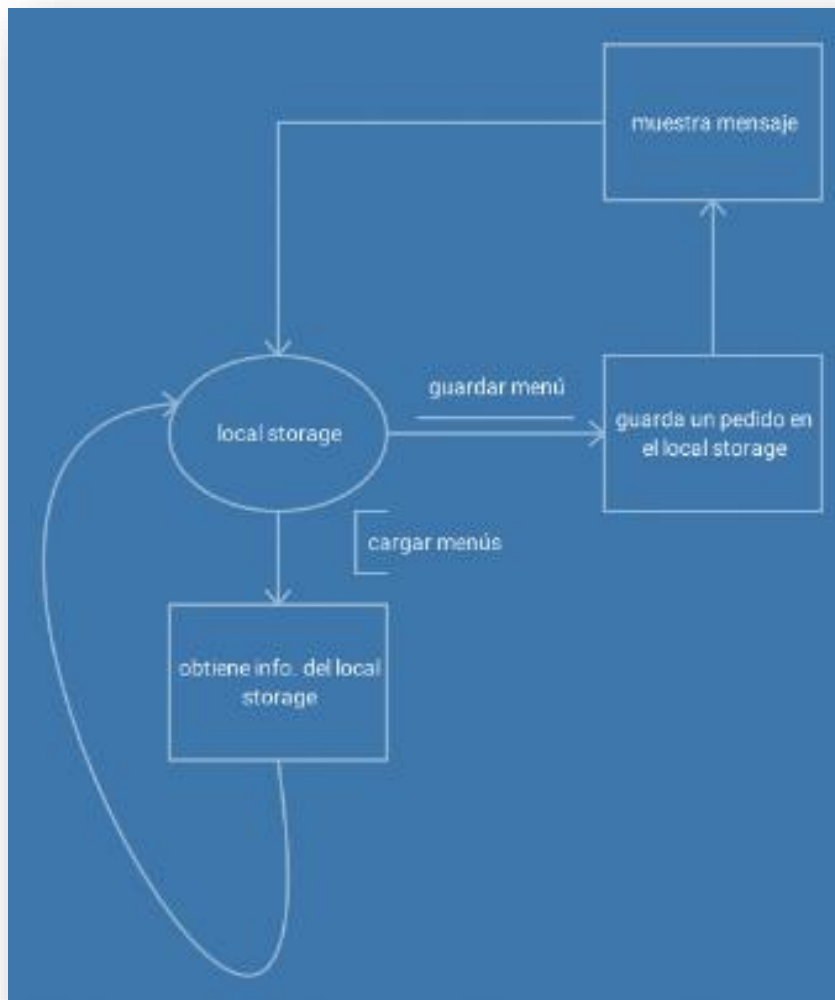


Figura 9. Diagrama de flujo de la app menú sección local storage.

APLICACIÓN COCINA

En el diseño de la aplicación móvil de cocina para un restaurante se establecieron en primera instancia las necesidades enlistadas a continuación:

El usuario de cocina visualiza los pedidos que recientemente se ingresaron, los pedidos en proceso de espera, y, los pedidos que están siendo preparados.

Cuando se da clic en el pedido que se encuentra en la lista de pedidos en espera, se despliega un recuadro emergente con los productos que se encuentran en el pedido, así como la información general. Luego, un botón para tomar el pedido, posteriormente, pasarlo a la lista de pedidos tomados y cocinando, como se muestra en la figura 9.

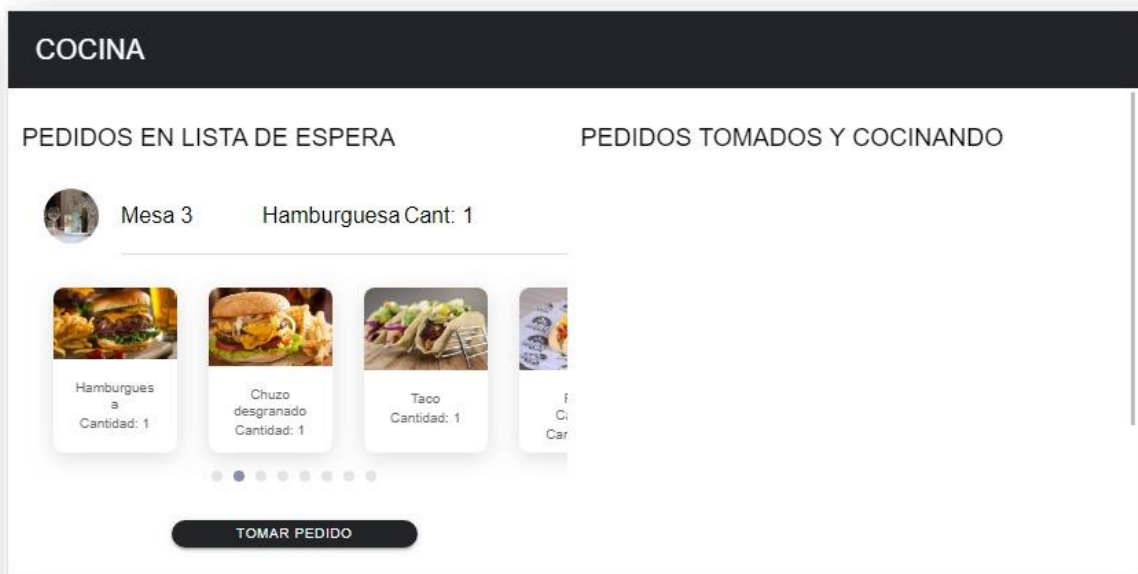


Figura 10. Pantalla de los pedidos mostrados en la cocina aplicación cocina.

En la aplicación móvil de cocina para un restaurante se muestran los pedidos que se encuentran en la lista de pedidos tomados y cocinando en la figura 9, a continuación, se enlistan los pasos de la funcionalidad:

Después que el usuario envié un pedido a la lista de pedidos tomados, en la parte derecha de la pantalla de la aplicación cocina se visualiza una lista desplegable que se compone de los productos ordenados por el usuario. Dichos productos ordenados son aquellos que el cocinero se encuentra preparando, como se muestra en la figura 9.

Al hacer clic en alguno de la lista de pedidos tomados, se despliega un recuadro con la información de los productos ordenados que se encuentran en el pedido actual, como se muestra en la figura 10.

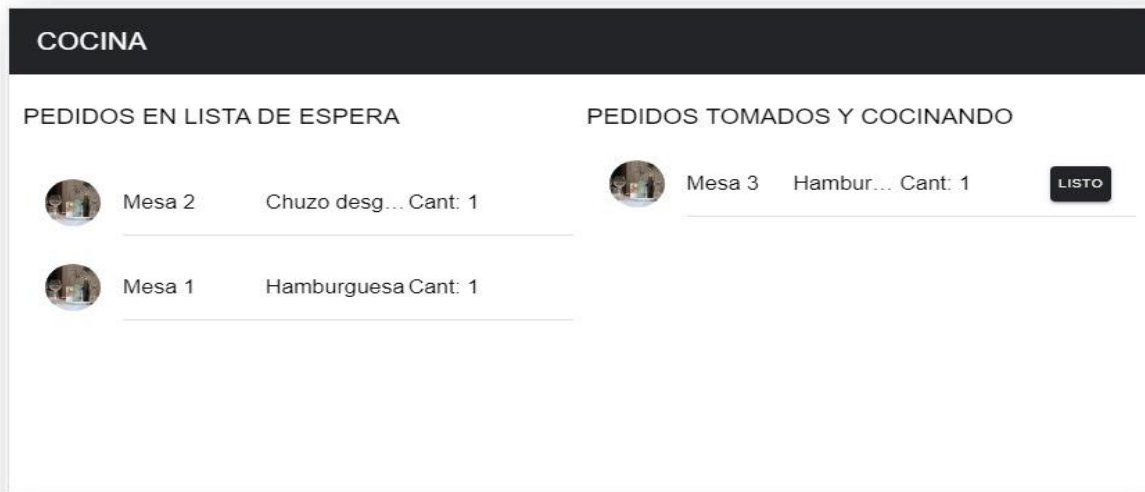


Figura 11. Pantalla de los pedidos mostrados en la cocina aplicación cocina.

En la aplicación móvil de cocina para un restaurante se muestran los pedidos que se encuentran en la lista de pedidos tomados y cocinado en la figura 10, a continuación, se enlistan los pasos de la culminación de las actividades a través del botón listo:

Una vez que en el área cocina terminen de preparar el pedido, puede proseguir a indicar al sistema la confirmación que la orden ya fue evacuada de la cocina. Para ello el usuario debe presionar el botón listo de la respectiva orden, en la lista de pedidos tomados y cocinando, como se muestra en la figura 11. Al confirmar un pedido con el botón listo la orden se deja de visualizar en la aplicación cocina, indicando que la orden ha sido evacuada del área de cocina.

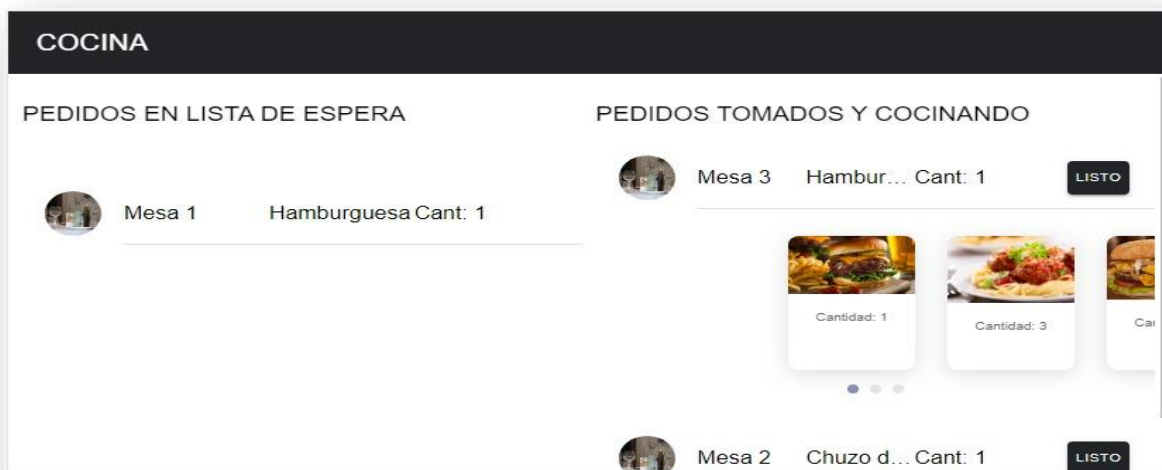


Figura 12. Pantalla de los pedidos mostrados en la cocina aplicación cocina.

El diagrama de flujo del sistema de home trabaja de la siguiente manera:

En la aplicación de cocina sólo hay una sección. Al Iniciar la aplicación, entra a la sección home, luego obtiene información de los menús y de los pedidos de firebase. Luego divide los pedidos en dos grupos, pedidos en espera y pedidos tomados y los visualiza. Luego comprueba si algún pedido en espera se selecciona, si no se selecciona, se devuelve a visualizar los pedidos en espera, si se selecciona, el pedido en espera seleccionado pasa a el grupo de pedidos tomados. Luego, comprueba si algún pedido tomado esta listo. Si no está listo, se devuelve a visualizar los pedidos tomados, si está listo, dicho pedido se elimina del grupo de pedidos tomados y pasa a estar listo. Por último, los pedidos listos salen de la sección de cocina, como se muestra en la figura 12.

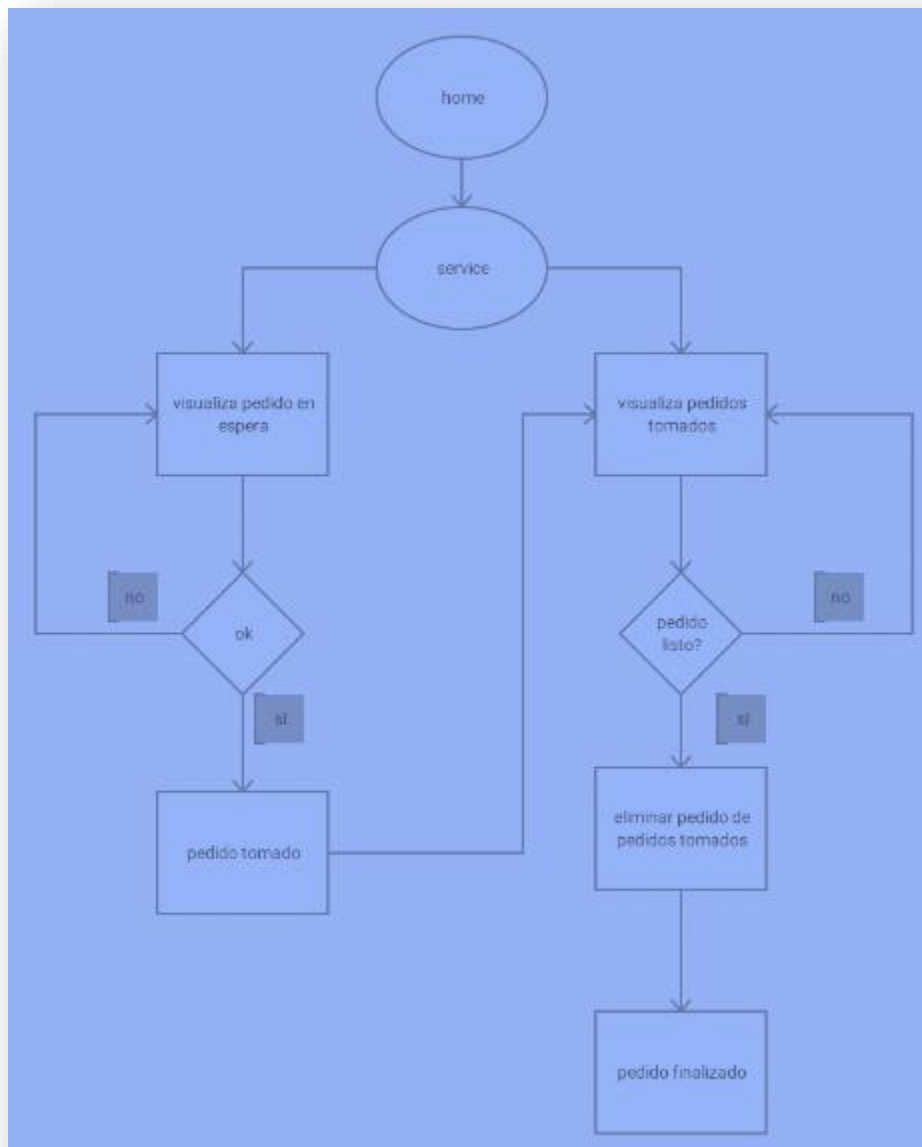


Figura 13. Diagrama de flujo de la app cocina sección home.

El diagrama de flujo del sistema de service trabaja de la siguiente manera:

La sección de service del programa de cocina tiene dos funciones, update pedido y load pedido. La función update Pedido actualiza la información de los pedidos del servidor firebase. La función load Pedido, obtiene la información de los pedidos existentes en el servidor firebase, como se muestra en la figura 13.

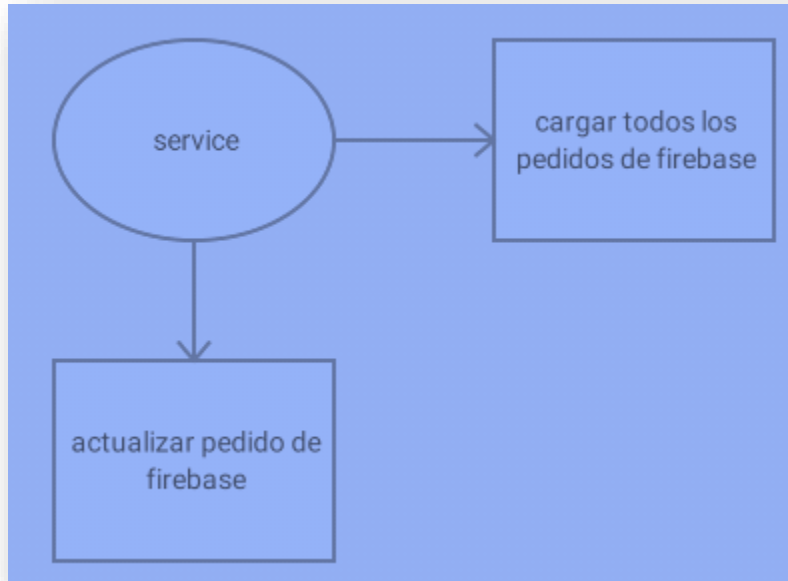


Figura 14. Diagrama de flujo de la app cocina de la sección service.

APLICATIVO WEB DE MONITOREO DE VENTAS, INVENTARIOS Y PEDIDOS

En la página principal del aplicativo web se encuentran el menú del aplicativo web y cada una de las mesas del restaurante.

En el menú del aplicativo se encuentran cada una de las diferentes ventanas, tales como, Home, Crear Mesa, Nuevo Producto y Menú. Cabe resaltar que este menú principal se encuentra en todo el aplicativo y en cada una de las diferentes ventanas del aplicativo web.

Como se puede observar en la figura 14, hay 3 Mesas, cada una en un estado diferente. La Mesa 3, se encuentra vacía y con un número de sillas 0, ya que no se encuentran clientes en dicha Mesa. En la Mesa 2 hay un número de sillas igual a 2, y un mensaje en color azul que indica el estado de la orden de dicha mesa, además, se observa un botón COBRAR de color Rojo, esto indica que la mesa está ocupada y en cualquier momento se puede dar la opción de cobrar la orden.

Cabe aclarar también que, el color del mensaje del estado de la mesa cambia dependiendo de su estado actual, naranja para cuando el pedido está a la espera de ser tomado por el cocinero, azul para cuando el pedido ha sido tomado en cocina y está siendo cocinado, y Verde para cuando el pedido está listo y ha sido entregado al cliente.

Si el usuario desea ver el estado de la mesa y los pedidos que se encuentran en la mesa, solo debe dar clic en la imagen o en el nombre de la mesa.

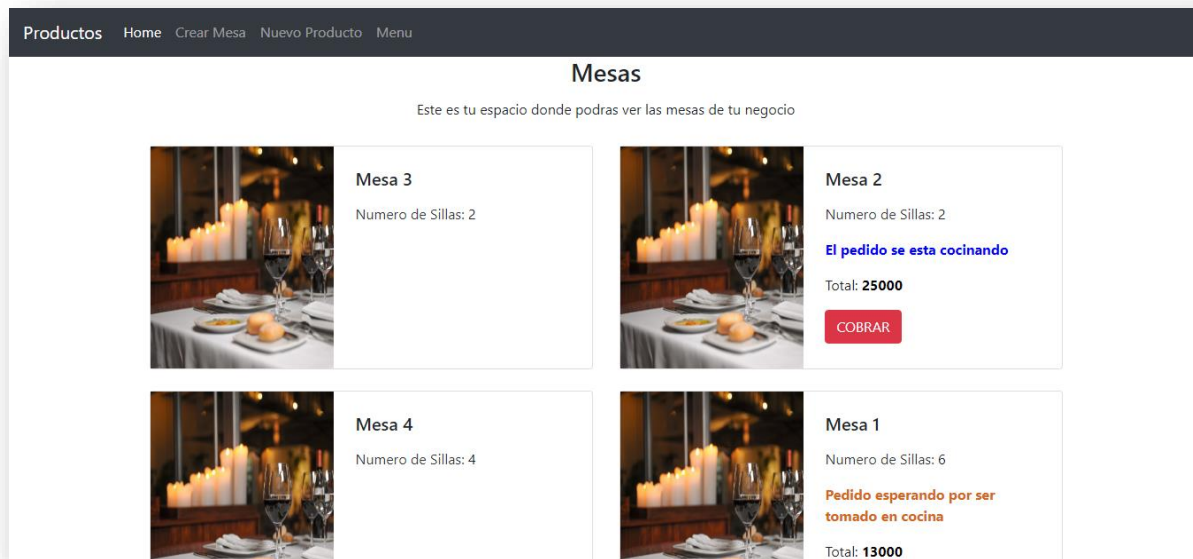


Figura 15. Ventana Home del aplicativo web

Al hacer clic sobre la imagen o el nombre de la mesa, el usuario será redirigido a la ventana de pedido de la mesa que selecciono.

La ventana de pedido se encuentra dividida por dos áreas. En el lado izquierdo se encuentran listados todos los productos existentes en el Menú, así como la información de nombre y precio. Además, para cada producto del menú también se encuentra un ítem numérico para ingresar la cantidad de productos que el usuario desee ordenar y un botón Add. Si el usuario hace clic sobre el botón Add, el producto respectivo será ingresado a la lista de productos del pedido.

En la parte derecha de la ventana de pedido se encuentran listados los productos del menú añadidos al

pedido. En esta lista se visualiza el nombre del producto, precio del producto y cantidad que el usuario anteriormente ingreso.

En la parte inferior derecha se encuentra el botón Realizar Pedido tal y como se observa en la Figura 15.

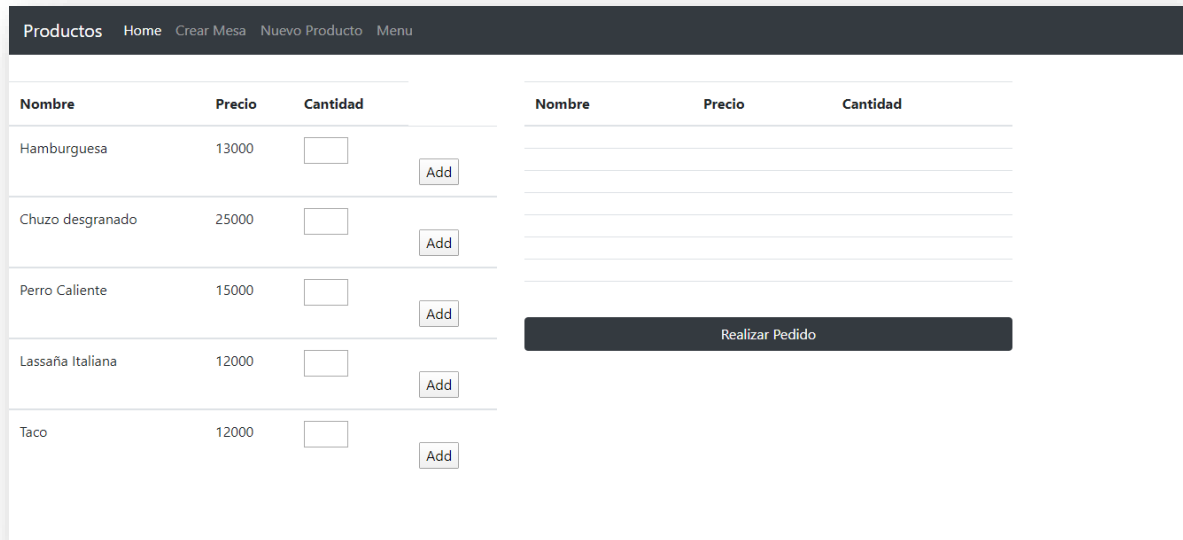


Figura 16. Ventana pedida del aplicativo web

Si el usuario no ha ingresado ningún producto del menú al pedido y hace clic sobre el botón Realiza Pedido se desplegará un mensaje avisando al usuario que por favor agregue un producto al pedido, tal y como se observa en la Figura 16.

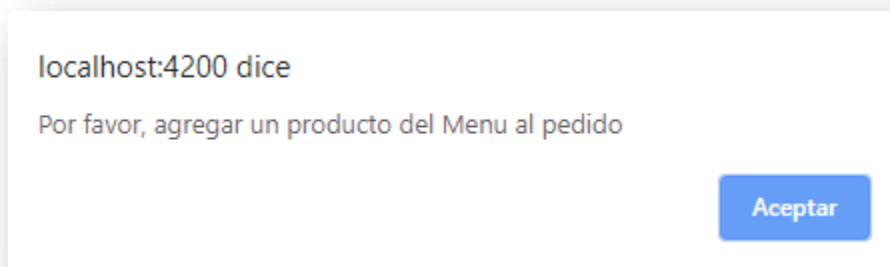


Figura 17. Mensaje de alerta pedido

Si el usuario ingreso algún producto al pedido y hace clic sobre el botón Realizar Pedido se desplegará un mensaje avisando al usuario si está seguro de realizar el pedido. En caso de confirmar, el pedido realizado será ingresado a la base de datos firebase y, por consiguiente, a todo el sistema, tal y como se observa en la Figura 17

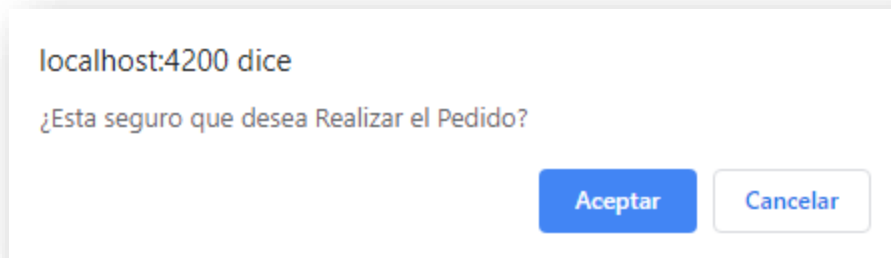


Figura 18. Mensaje de confirmación pedido

La ventana de crear mesa se encuentra dividida por dos áreas. En la parte izquierda se encuentra el recuadro donde el usuario puede ingresar el Número de la mesa y el número de Sillas. Luego de ingresados los datos de la nueva mesa, el usuario podrá hacer clic en el botón GUARDAR para crear la nueva mesa en el sistema.

Como se puede observar en la Figura 18, en la parte derecha de la ventana de Crear Mesa se encuentran listadas cada una de las mesas existentes. En esta área de la ventana se puede visualizar el nombre de la Mesa, Numero de sillas y además, se encuentra un botón Borrar Mesa, en donde el usuario puede eliminar la mesa correspondiendo haciendo clic en el botón Borrar Mesa.

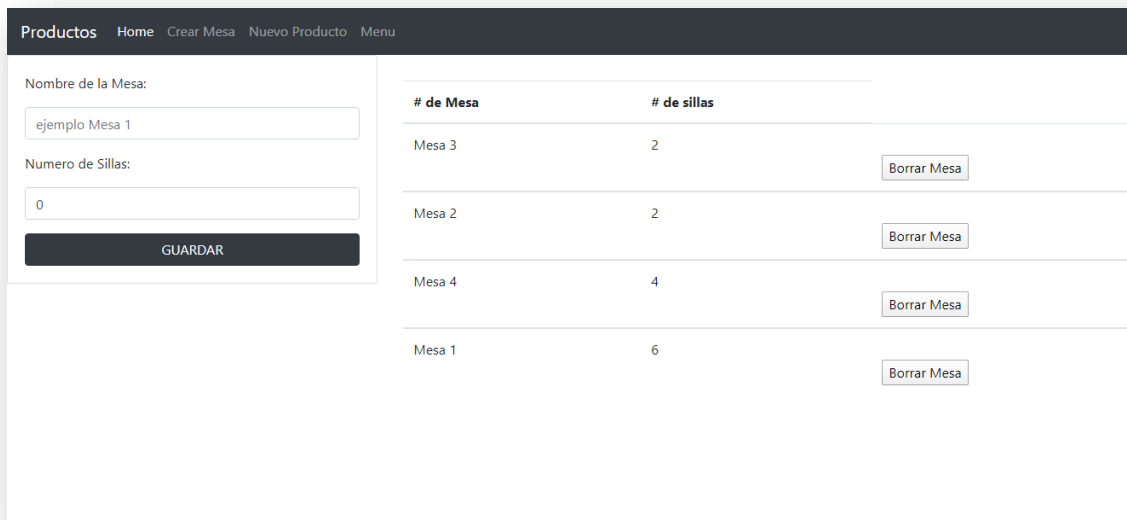


Figura 19. Ventana Crear Mesa del aplicativo web

Si el usuario da clic en el botón Borrar Mesa se despliega un mensaje de alerta en la parte superior de la ventana. En el mensaje de alerta se le pregunta al usuario si desea Eliminar la Mesa, y tiene la opción de confirmar o cancelar la acción de eliminar mesa tal y como se observa en la Figura 19.

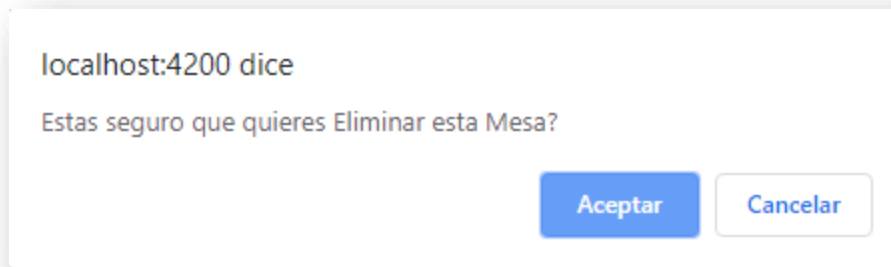


Figura 20. Mensaje de confirmación crear mesa

La ventana de Nuevo Producto se encuentra dividida en dos áreas, en la parte izquierda de la ventana se encuentra un recuadro con una lista de ítems. Estos ítems son entradas de texto y numéricos, en lo cual se le solicita al usuario que ingrese los datos de Nombre del producto, descripción del producto, etiqueta del producto, precio del producto y cantidad de productos a ingresar. Para crear un nuevo producto en el inventario el usuario puede hacer clic en el botón SAVE que se encuentra ubicado en la parte inferior izquierda de la ventana.

En la parte derecha de la ventana de Nuevo producto se encuentra listado el inventario existente en el servidor firebase. En esta lista se puede observar el nombre, precio y cantidad de productos existentes en el inventario. Además, se encuentra un botón Editar en la parte derecha de cada producto listado, tal y como se observa en la Figura 20.

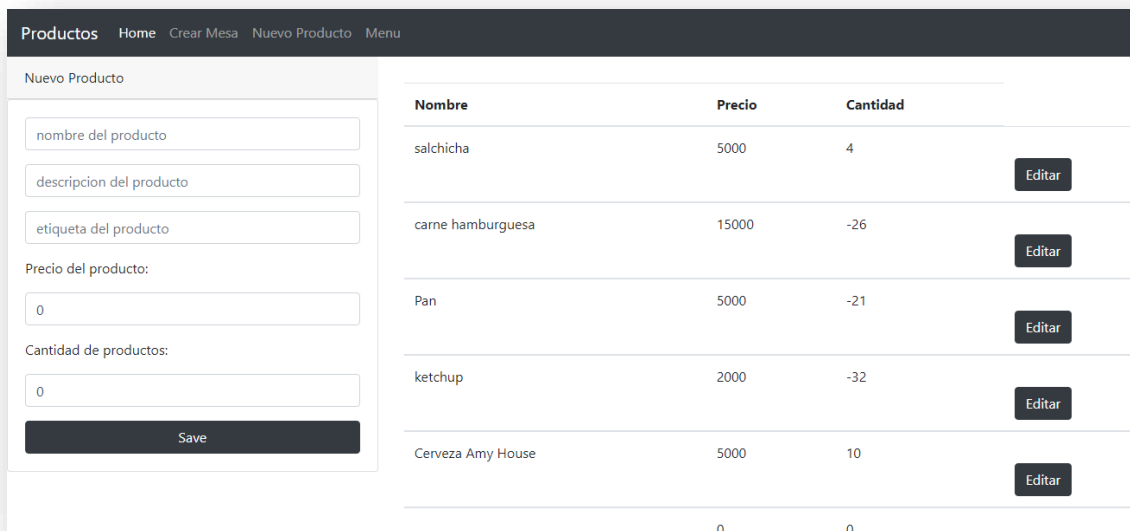


Figura 21. Ventana Nuevo producto del aplicativo web.

Si el usuario hace clic en el botón Editar será redirigido a una nueva ventana, en la cual, el usuario puede editar el producto seleccionado en la ventana de Nuevo producto. Tal y como se observa en la Figura 21, en la ventana de Editar producto se encuentra una lista de entradas de texto que ocupan el total del ancho de la ventana. En estas entradas de texto y número el usuario puede editar los campos de Nombre del producto, descripción de producto, Etiqueta del producto, Precio del producto, cantidad de productos y productos vendidos.

En la parte inferior de la lista, se puede observar un botón Modificar, el cual, si el usuario hace clic sobre él, la información ingresada por el usuario será modificada en la base de datos firebase y será redirigido a la ventana Nuevo producto.

The screenshot shows a web application interface for editing a product. At the top, there is a navigation bar with the following items: "Productos", "Home", "Crear Mesa", "Nuevo Producto", and "Menu". Below the navigation bar, the page title is "Editar Producto". The form consists of several input fields:

Nombre del producto:	salchicha
Descripción del producto:	salchicha texana para perror calientes
Etiqueta del producto:	carnes
Precio del producto:	5000
Cantidad de productos:	4
Productos vendidos:	6

At the bottom of the form, there is a dark button labeled "Modificar".

Figura 22. Ventana Editar producto del inventario del aplicativo web

La ventana Menú se encuentra dividida en dos áreas como se puede observar en la Figura 22, en la parte izquierda se encuentra un recuadro con una lista de ítems a ingresar. En estos ítems el usuario puede ingresar los atributos de Nombre del producto, descripción del producto, etiqueta del producto y precio del producto. Además de los anteriormente mencionados, se encuentra un ítem llamado Ingredientes, en este ítem el usuario puede seleccionar uno o varias opciones listadas. Las opciones listadas en el ítem ingredientes son los diferentes productos existentes en el inventario.

En la parte inferior del lado izquierda de la ventana Menú, se encuentra un botón Save, si el usuario hace clic sobre este botón, toda la información ingresada por el usuario se guardará en la base de datos firebase.

En la parte derecha de la ventana Menú se encuentra listados todos los productos existentes en el Menú. Tal y como se observar en la Figura 22, los datos mostrados de cada producto son el nombre, precio y etiqueta.

Productos Home Crear Mesa Nuevo Producto Menu

Nuevo Producto del Menu

nombre del producto

descripcion del producto

etiqueta del producto

Ingredientes

- salchicha
- carne hamburguesa
- Pan
- ketchup

0

Save

Nombre	Precio	Etiqueta
Hamburguesa	13000	Comida Rapida
Chuzo desgranado	25000	Comida Rapida
Perro Caliente	15000	Comida Rapida
Taco	12000	Comida Rapida

Figura 23. Ventana Menú del aplicativo web.

El diagrama de flujo del sistema de home trabaja de la siguiente manera:

La sección home del aplicativo web primeramente obtiene la información de las mesas existentes de firebase. Luego visualiza los pedidos en cada una de las esas y comprueba si la mesa está vacía. Si está vacía, vuelve a visualizar los pedidos, sino está vacía, muestra el botón cobrar. En la sección home del aplicativo web hay 4 funciones, crear mesa, nuevo producto, menú y nuevo pedido, como se muestra en la figura 23.

- La función crear mesa direcciona a la sección crear mesa.
- La función nuevo producto direcciona a la sección inventario.
- La función menú direcciona a la sección menú.
- La función nueva pedido direcciona a la sección pedido.

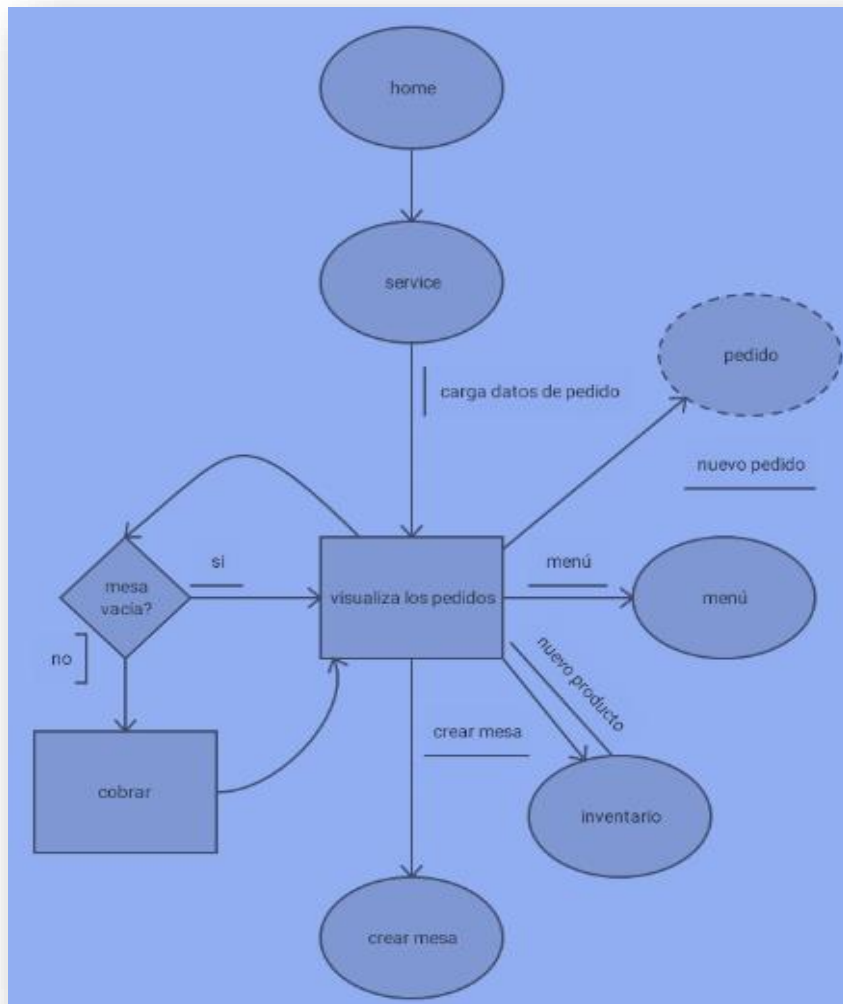


Figura 24. Diagrama de flujo de la app web de la sección home.

El diagrama de flujo del sistema de crear mesa trabaja de la siguiente manera:

La sección crear mesa primeramente llama a la sección service y obtiene la información de las mesas existentes. Después de obtenida la información de las mesas existentes, visualiza las mesas y visualiza la parte de ingresar una nueva mesa. Después de ingresar nueva mesa, llama a la sección service y crea una nueva mesa en el servidor firebase, luego vuelve a la sección home. En la parte de visualizar las mesas existentes, si se desea borrar la mesa, se comprueba si la mesa está ocupada. Si la mesa está ocupada, se muestra un mensaje diciendo que la mesa no se puede borrar porque está ocupada, si la mesa no está ocupada llama al services para borrar la mesa del servidor firebase y, luego se devuelve a visualizar las mesas existentes, como se muestra en la figura 24.

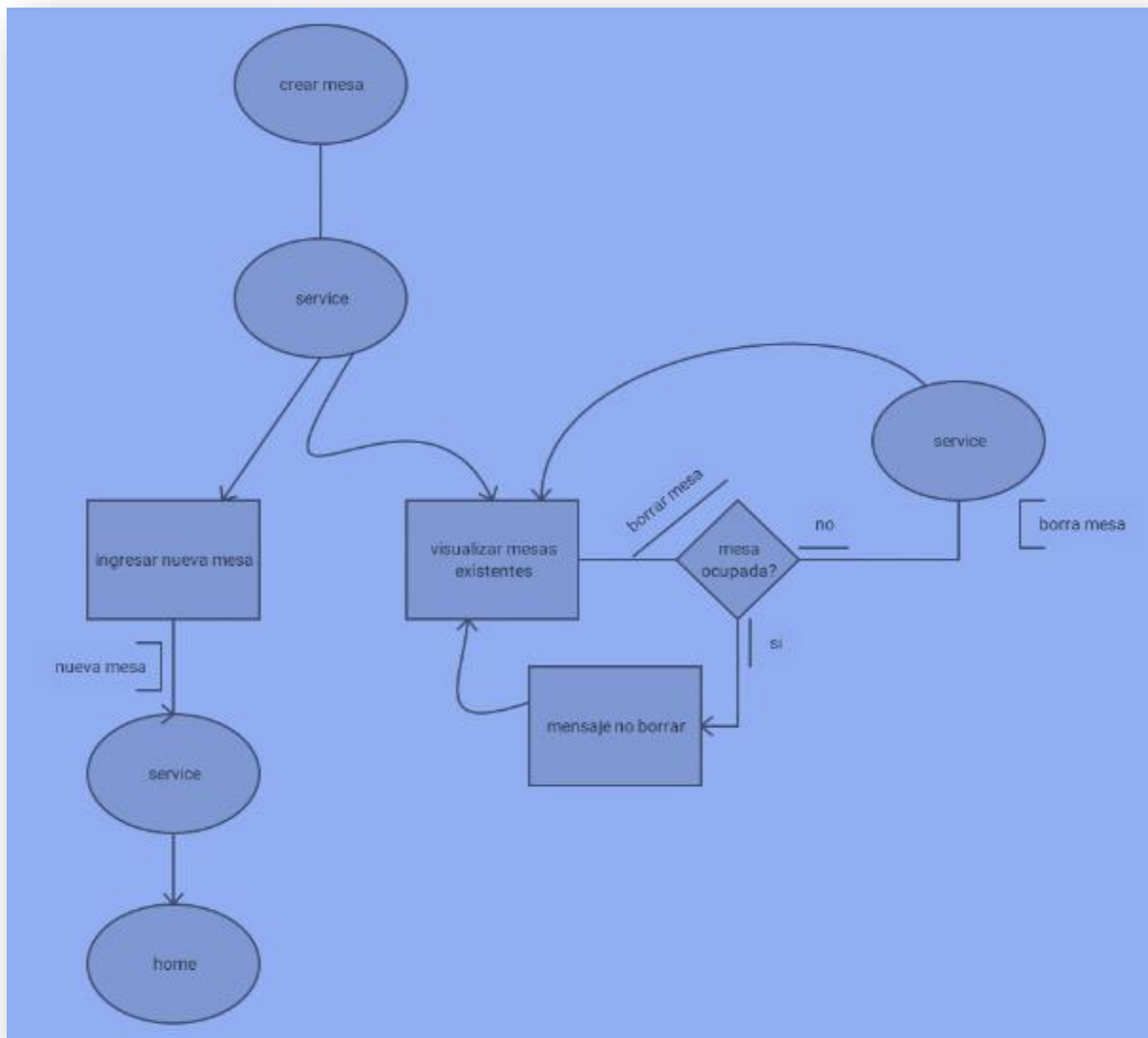


Figura 25. Diagrama de flujo de la app web de la sección creación de mesa.

El diagrama de flujo del sistema de inventario trabaja de la siguiente manera:

Inicialmente, en la sección inventario se llama al service y se obtiene la información del inventario existente en la base de datos de firebase. Después de obtenida la información del service, se pasa a visualizar el inventario y agregar nuevo producto. En la parte de visualizar inventario, si se solicita editar producto, se edita y se llama al service para guardar la actualización del producto en la base de datos de firebase. En la parte de agregar nuevo producto, primeramente, se comprueba si el producto ya existe. Si el producto agregado existe, se muestra un mensaje informando que el producto nuevo ya existe y se regresa a ingresar nuevo producto. Si el producto nuevo no existe, se llama al service y se crea el nuevo producto en la base de datos de firebase, como se muestra en la figura 25.

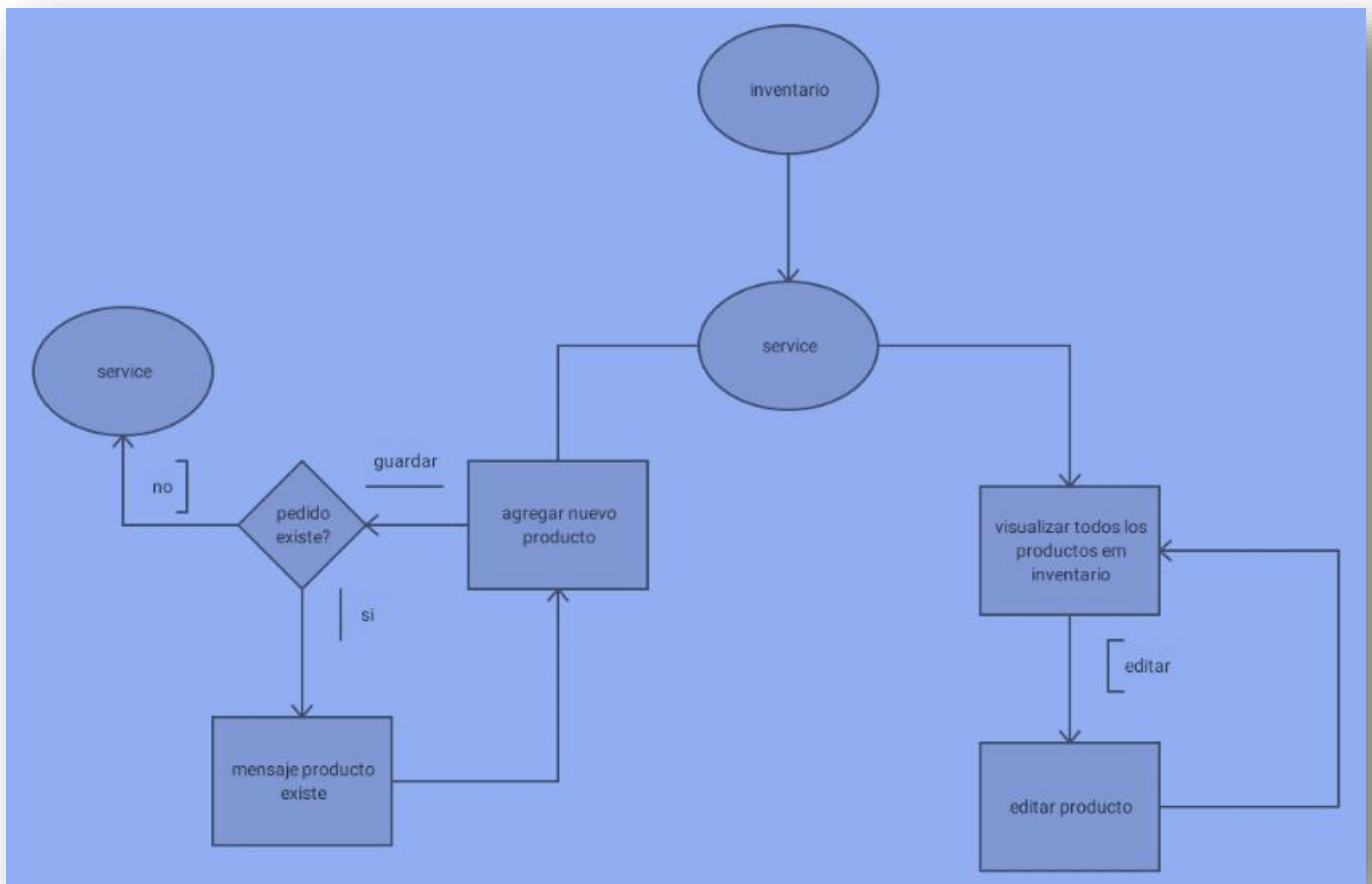


Figura 26. Diagrama de flujo de la app web sección inventario.

El diagrama de flujo del sistema de menú trabaja de la siguiente manera:

En la sección menú como primera medida se llama al service y se obtiene información del menú de la base de datos de firebase. Luego de obtenida la información del menú, se visualizan los menús y se inicializa la parte de agregar nuevo producto en menú. Si se agrega un nuevo producto en menú, se comprueba si el producto existe en el menú. Si existe, se muestra un mensaje indicando que el producto ya existe. Si no existe, se llama al service y se guarda el nuevo producto en la base de datos del service, como se muestra en la figura 26.

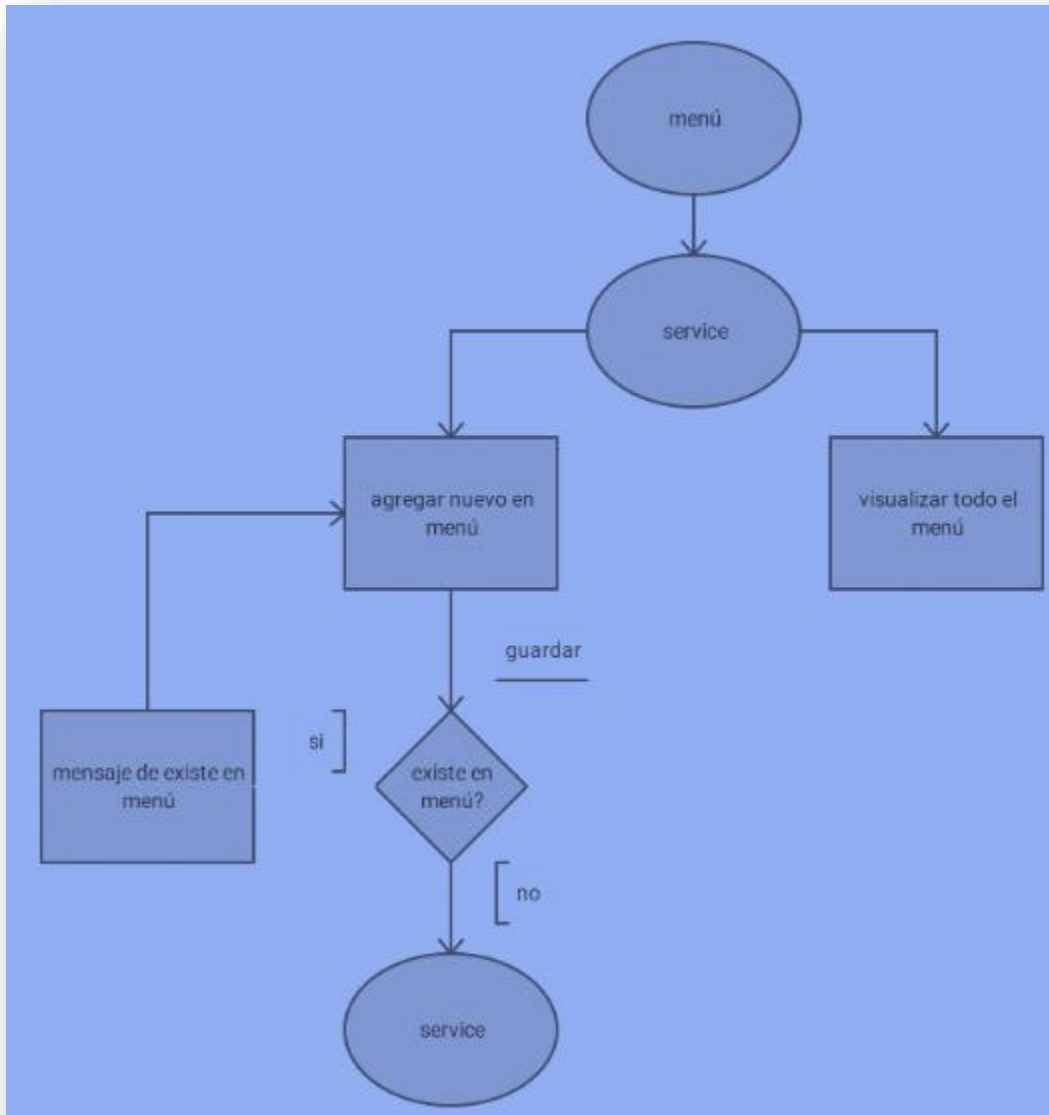


Figura 27. Diagrama de flujo de la app web sección menú.

El diagrama de flujo del sistema de pedido trabaja de la siguiente manera:

Al ingresar en la sección de pedido se llama al service, y carga la información de pedidos y menú que están alojados en el servidor de firebase. Después, existen dos flujos de información, en la primera hay visualización del menú, y en el segundo, el flujo de la organización de pedidos a ordenar.

En la visualización del menú se muestran los productos que están en el menú. Al realizar la función añadir se comprueba si la cantidad es diferente de 0. Si es diferente de 0, entonces, el producto añadido se envía al flujo de la organización de pedidos a ordenar.

Al inicializar la función realizar pedido, se comprueba si hay productos a ordenar, si no hay productos a ordenar, el sistema regresa a el flujo de organización de pedidos a ordenar. Así que, si existen productos a ordenar se llama al service y se guarda el pedido en el servidor de firebase, como se muestra en la figura 27.

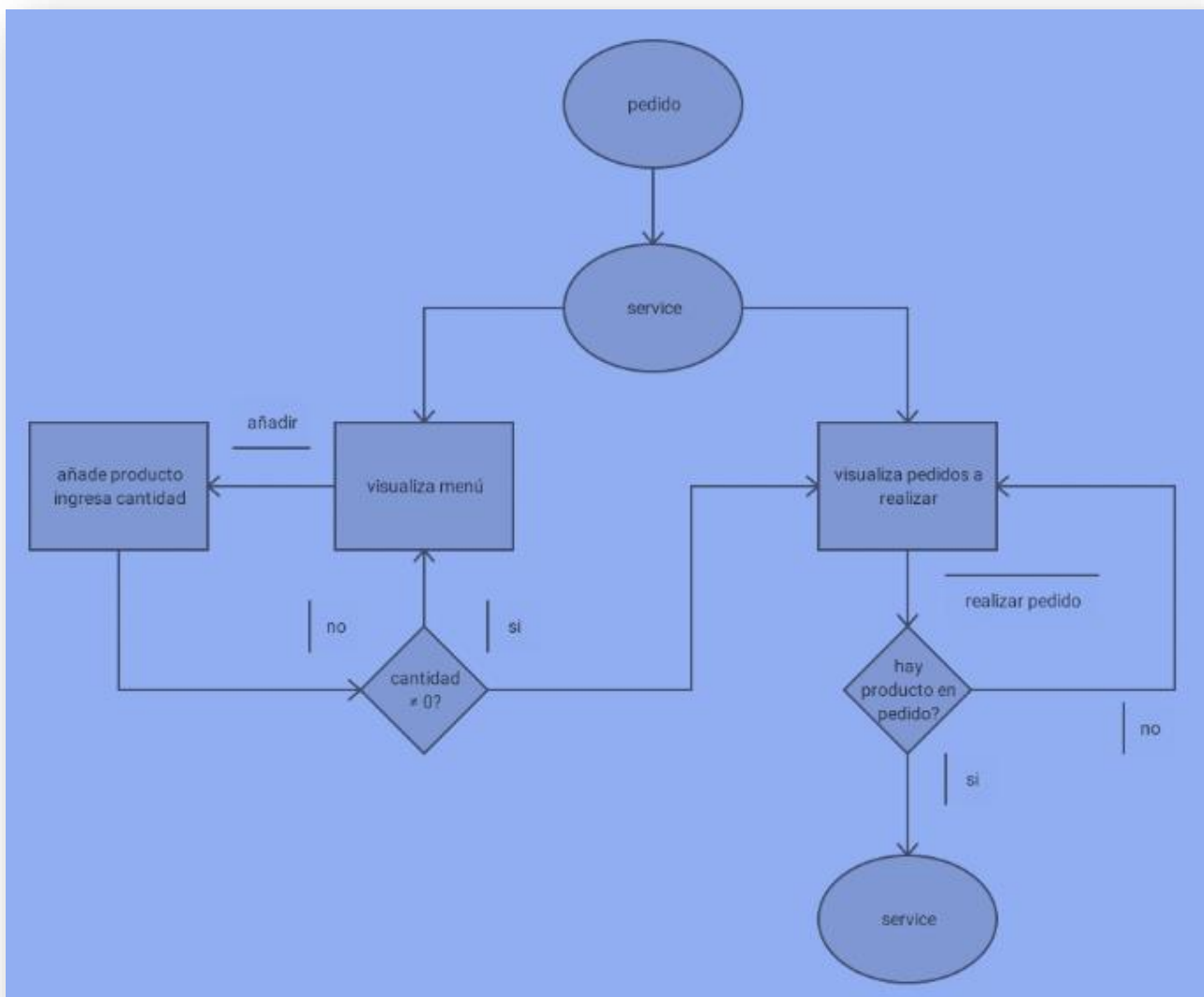


Figura 28. Diagrama de flujo de la app web sección pedido.

El diagrama de flujo del sistema de cobro trabaja de la siguiente manera:

Primero, se debe llamar a la sección service retornando de esta forma la información de menús, pedidos e inventario que se encuentran alojados en el servidor de firebase. Luego, se compara si el pedido es igual al menú, si no es así el sistema retorna a la comprobación. Entonces, si el pedido es igual al menú se procede a hacer la comprobación del producto con el inventario, porque es necesario conocer qué tipo de ingredientes tiene el producto que se ordenó. Si los ingredientes del inventario no coinciden con el producto, el sistema retorna la comprobación, pero si el producto corresponde a los ingredientes del inventario se prosigue a modificar los campos de cantidad y vendidos del documento inventario en el servidor firebase.

Por último, se reinician todos los campos del pedido y se llama a la sección service el cual actualiza la información de pedido e inventario en el servidor firebase, como se muestra en la figura 28.

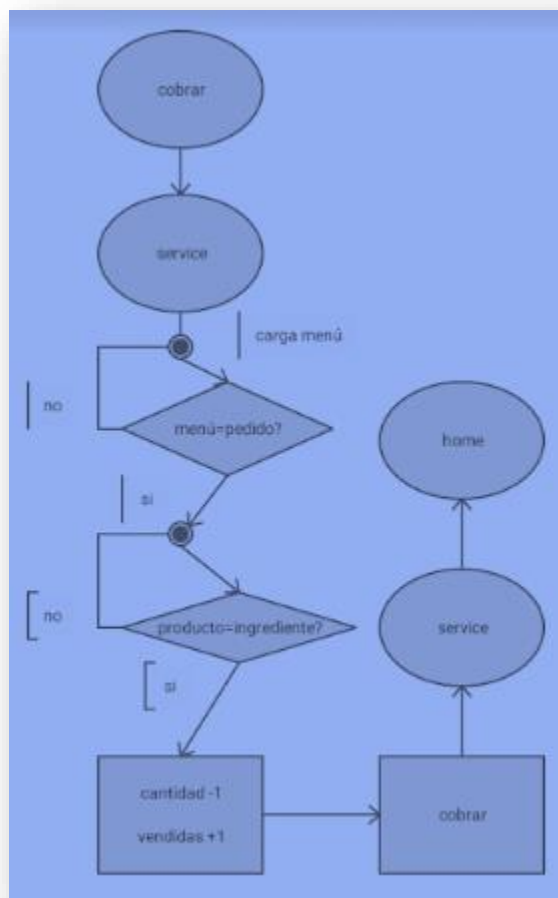


Figura 29. Diagrama de flujo de la app web de la sección sistema de cobro.

El diagrama de flujo de la sección service como se muestra en la figura 29 trabaja de la siguiente manera:

Cuando la sección de service es llamada, el service procede a realizar una de sus funciones, las cuales se sincronizan con la base de datos firebase. Después de realizar la conexión con firebase el sistema retorna a la sección que llamo al service anteriormente.

- **Borrar pedido de firebase:** elimina un elemento de la colección de pedidos en el servidor firebase.
- **Borrar producto de firebase:** elimina un elemento de la colección de productos en el servidor firebase
- **Borrar menú de firebase:** elimina un elemento de la colección de menú en el servidor firebase
- **Actualizar pedido de firebase:** actualiza un elemento de la colección de pedidos en el servidor firebase.
- **Actualizar producto de firebase:** actualiza un elemento de la colección de producto en el servidor firebase.
- **Actualizar menú de firebase:** actualiza un elemento de la colección de menú en el servidor firebase.
- **Agregar producto:** crea un nuevo ítem en la colección de productos en el servidor firebase.
- **Agregar menú en firebase:** crea un nuevo ítem en la colección menú en el servidor firebase.
- **Agregar nuevo pedido en firebase:** crea un nuevo ítem en la colección pedidos en el servidor firebase.
- **Cargar todos los productos de firebase:** carga todos los ítems de la colección productos en el servidor firebase.
- **Cargar todos los menús de firebase:** carga todos los ítems de la colección menú en el servidor firebase.
- **Cargar todos los pedidos de firebase:** carga todos los ítems de la colección pedidos en el servidor firebase.
- **Cargar un producto de firebase:** carga un ítem de la colección productos en el servidor firebase.
- **Cargar un pedido de firebase:** carga un ítem de la colección pedidos en el servidor firebase.
- **Cargar un menú de firebase:** carga un ítem de la colección menú en el servidor firebase.

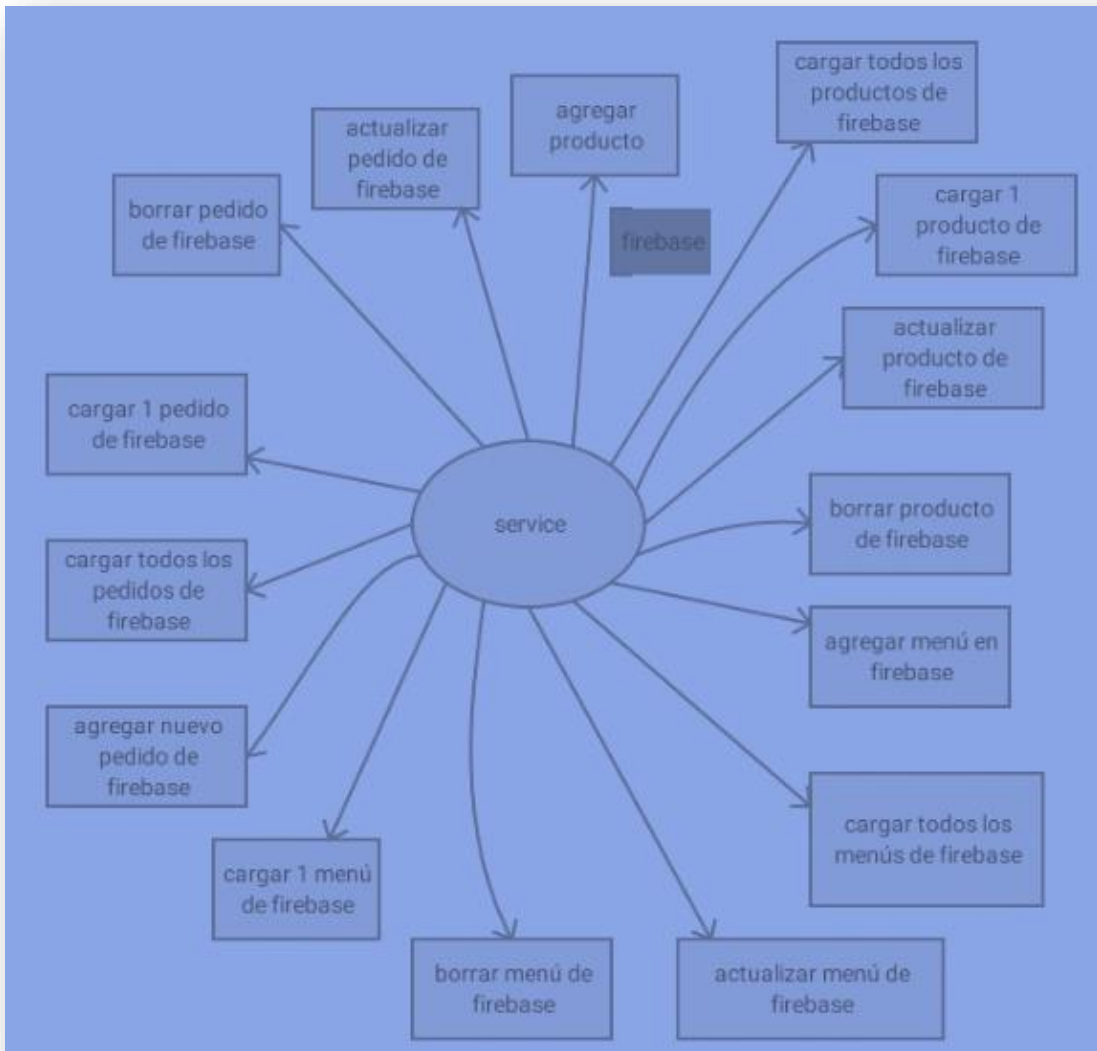


Figura 30. Diagrama de flujo de la app web sección service.

INTEGRACIÓN DE LOS DOS APLICATIVOS MOVILES Y EL APLICATIVO WEB EN UN ÚNICO SISTEMAS

En la figura 30 se muestran las pantallas iniciales del aplicativo menú, del aplicativo de la cocina y del aplicativo web respectivamente en secuencias.

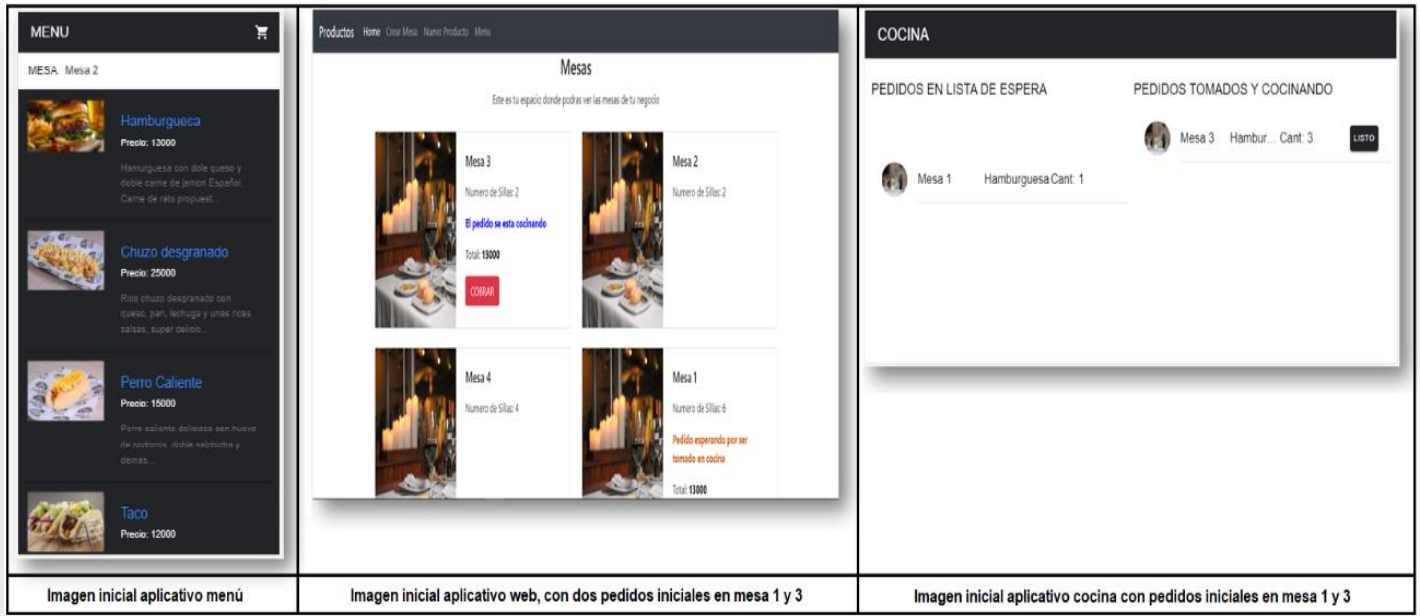


Figura 31. Pantallas iniciales de los aplicativos menú, cocina y aplicativo web.

En la figura 31 se muestra la secuencia de cómo se debe realizar un pedido en el aplicativo móvil de menú, en donde, inicialmente en la página principal de la app lo que se debe hacer es seleccionar un

pedido enlistado cliqueando en la etiqueta del nombre del producto. Posteriormente, el vínculo cliqueado lleva a una nueva pantalla que muestra la descripción del producto completamente detallada con las características de precio, descripción de los ingredientes y una imagen explicativa. Entonces, el usuario podrá agregar el producto mostrado al carrito de compras, cuando es agregado el producto al carrito de compras aparece una ventana emergente de confirmación del pedido para anexar finalmente el producto a la lista de pedidos.

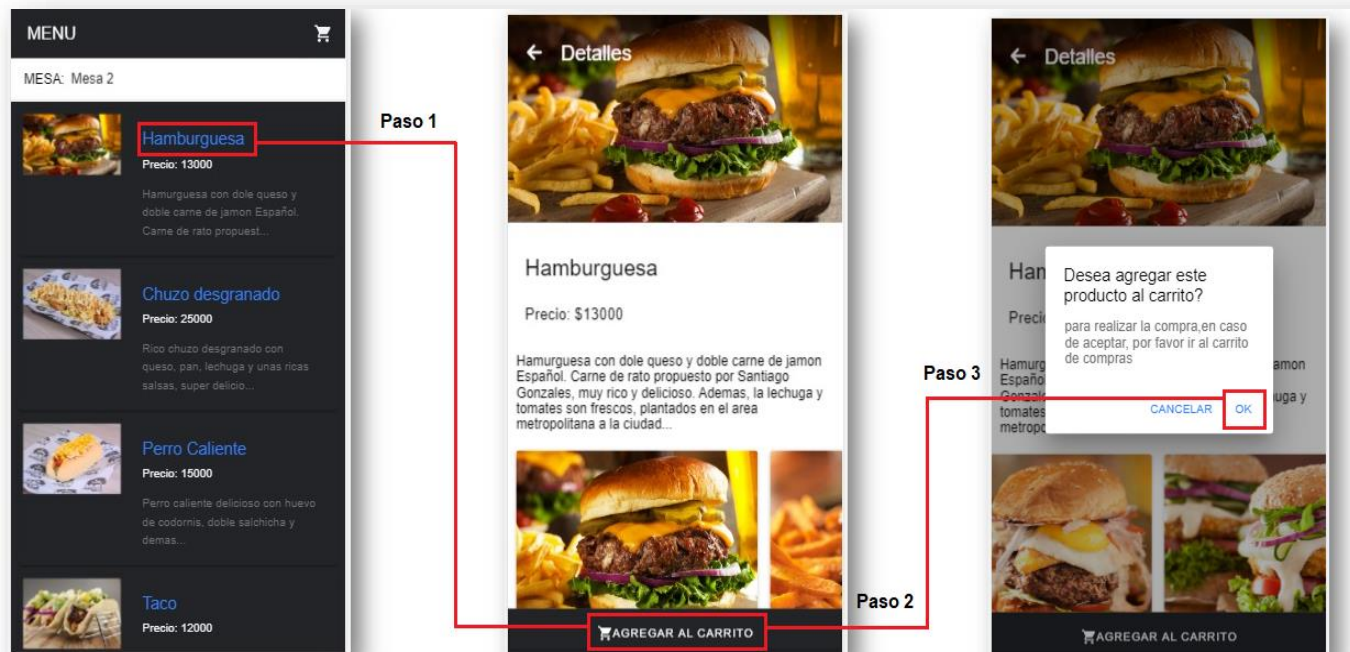


Figura 32. Inicio del pedido en el aplicativo móvil de menú.

En la figura 32 se muestra la secuencia del pedido ya agregado a la sección carrito de pedido, se puede apreciar una ventana con el ítem escogido por el usuario, entonces, es importante cliquer una pequeña casilla de selección que le permite al usuario tomar el pedido para comprarlo y luego se mostrará una pequeña ventana emergente de confirmación para asegurar la compra, incluso si se desea en el icono de basura se puede eliminar la selección y regresar al menú principal a tomar nuevamente la orden. En la ventana de compra se muestran diferentes características tales como el precio del ítem, la cantidad de ítems que se desean anexar, los botones de sumar y restar para hacer variaciones al ítem en cuanto a la cantidad y una casilla de selección de todo para poder tomar el número de pedidos totales que estén presentes en la lista de compra.

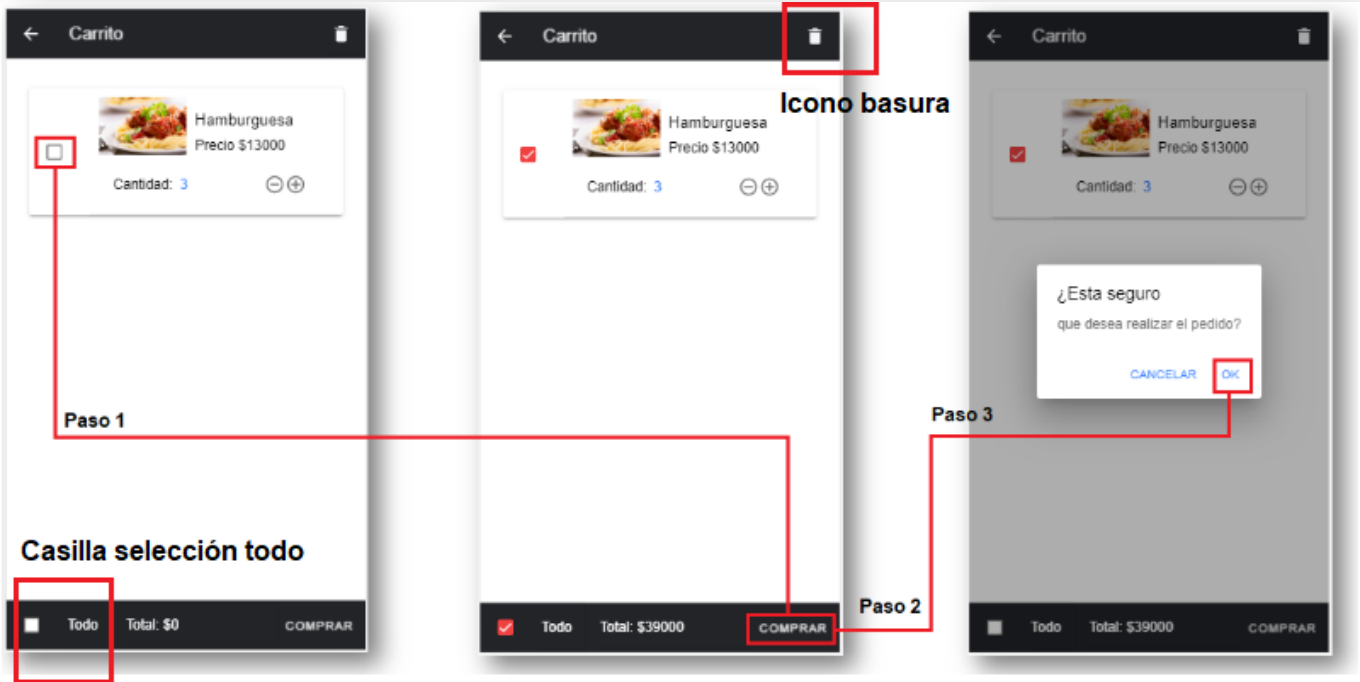


Figura 33. Secuencia del pedido en el aplicativo móvil de menú.

En la figura 33 se muestra en las pantallas del aplicativo de la cocina y del aplicativo web, el pedido que se ha asignado por el usuario en el aplicativo menú. En la imagen se puede ver el ejemplo de un pedido en la mesa 2, que a su vez se reflejado en los tres aplicativos al tiempo.

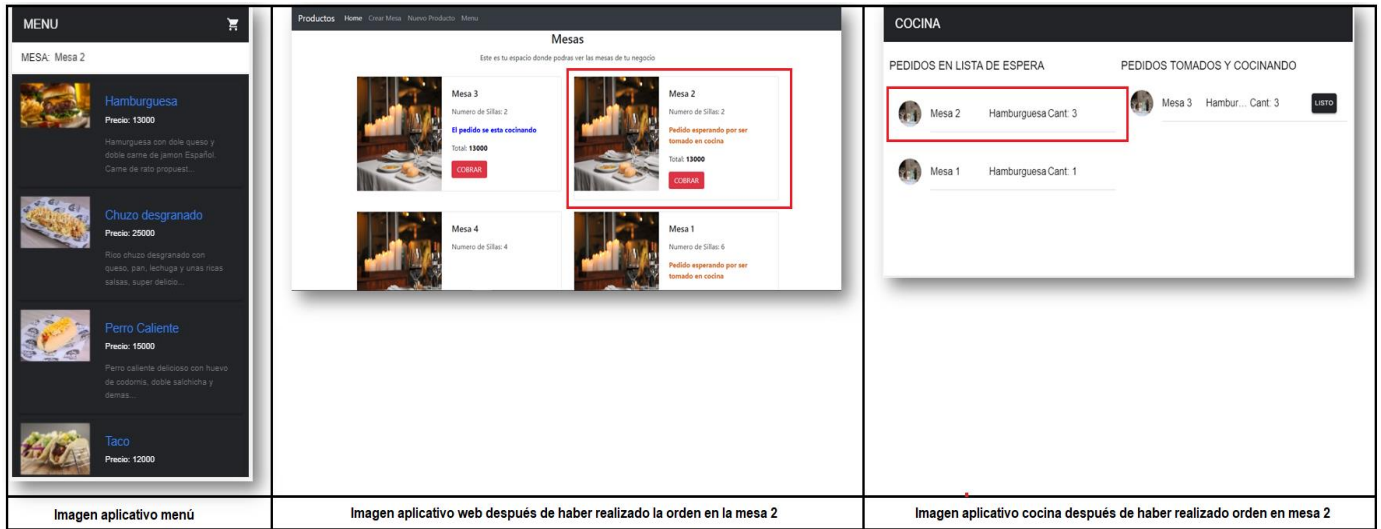
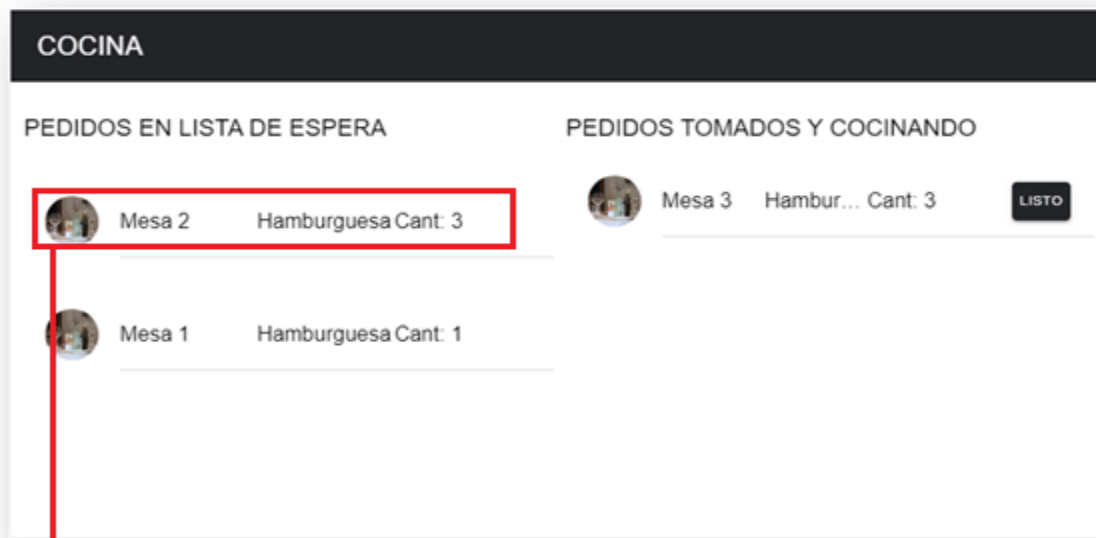
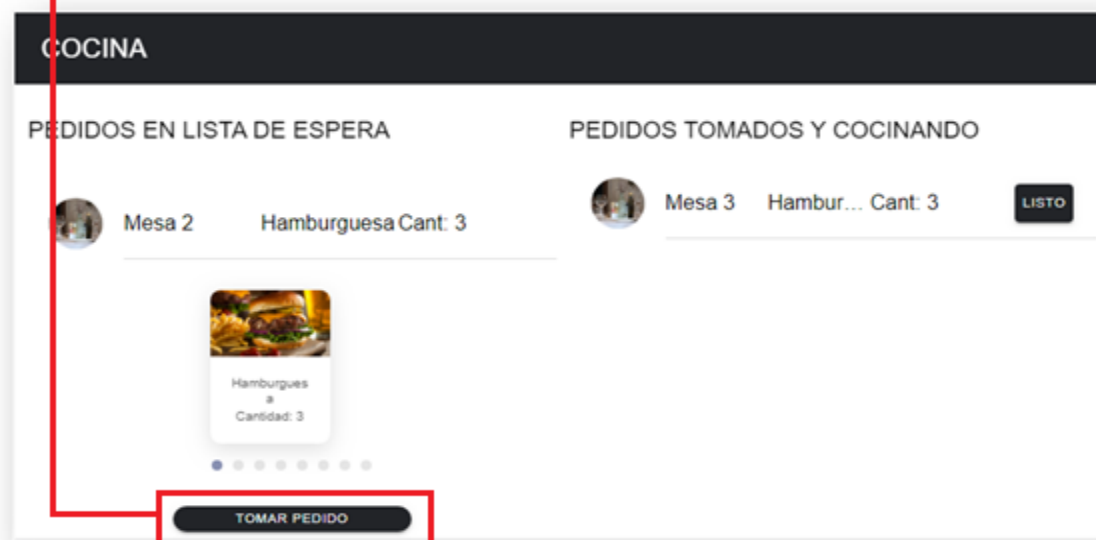


Figura 34. Pantallas de los aplicativos menú, cocina y aplicativo web, después del pedido en mesa 2.

En la figura 34 una vez realizado el pedido en la app menú se ve reflejado el pedido en la app cocina. Por lo tanto, es posible ver el pedido realizado por el usuario en la lista de pedidos en espera. Al clicar en el pedido mesa seleccionada en la lista de pedidos de espera, se despliega un recuadro con la información de los productos asignados al respectivo pedido. En consecuencia, se cliquea en el botón tomar pedido para indicar al sistema que el pedido se ha tomado y se prosigue a cocinar. De modo que el pedido pasa a la lista de pedido tomados y cocinando.



Paso 1



Paso 2

Figura 35. Secuencia pantallas del aplicativo de cocina para tomar el pedido.

En las figuras 35 y 36 se muestra el pedido en estado tomado y cocinando en los aplicativos de menú, cocina y el aplicativo web. Como se observa en la figura 35 el pedido en la mesa 2 indica por medio de un mensaje en color azul que el pedido ha sido tomado y se está cocinando.

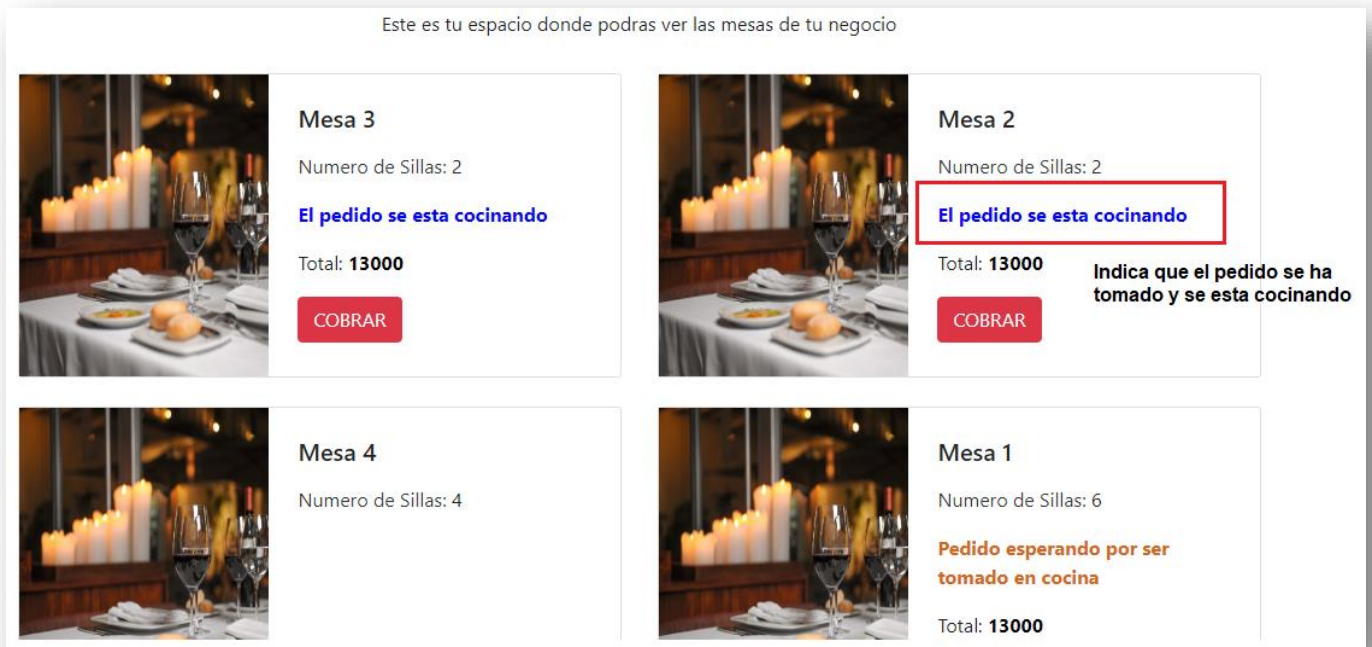


Figura 36. Indicación de pedido tomado y cocinando.

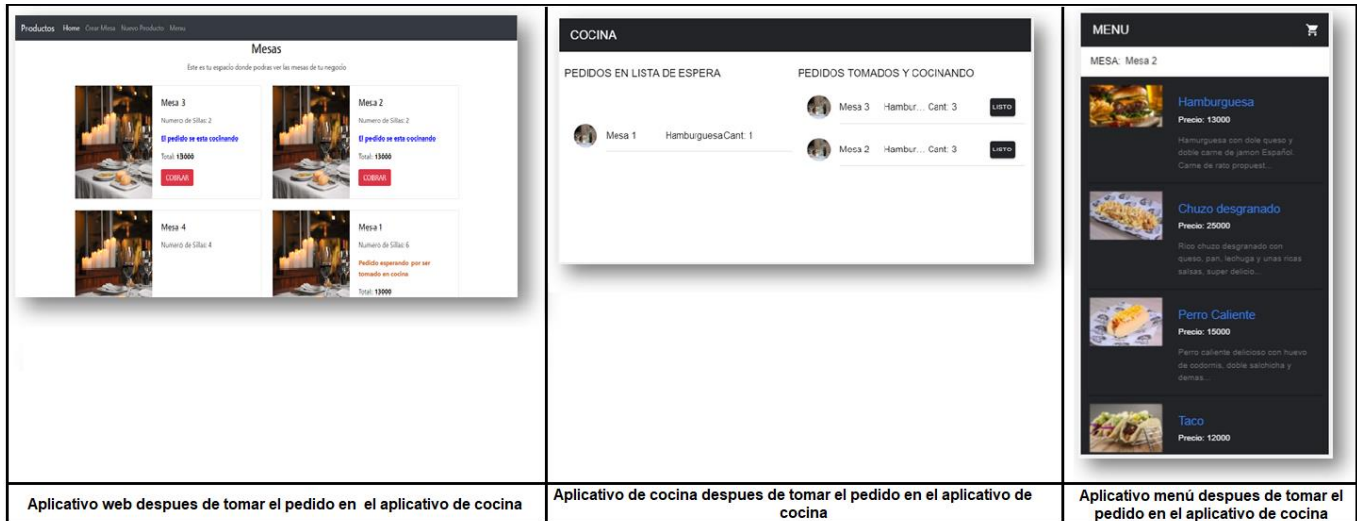


Figura 37. Secuencia pantallas del aplicativo de cocina para tomar el pedido.

Una vez el pedido se encuentre en la lista de pedidos tomados y cocinando, es necesario que se indique al sistema cuando la orden esta lista para servir a la mesa respectiva. En la figura 37 se observan los pasos para indicar al sistema cuando el pedido está listo. Primeramente, se debe dar clic en cualquier pedido de la lista de pedidos tomados y cocinando, después de esto, se desplegará un recuadro con la información de los productos contenidos en dicho pedido. En el momento que el usuario desee indicar que todos los productos se han cocinado y están listos, se debe dar clic en el botón LISTO del respectivo

pedido, tal y como se observa en la figura 37.

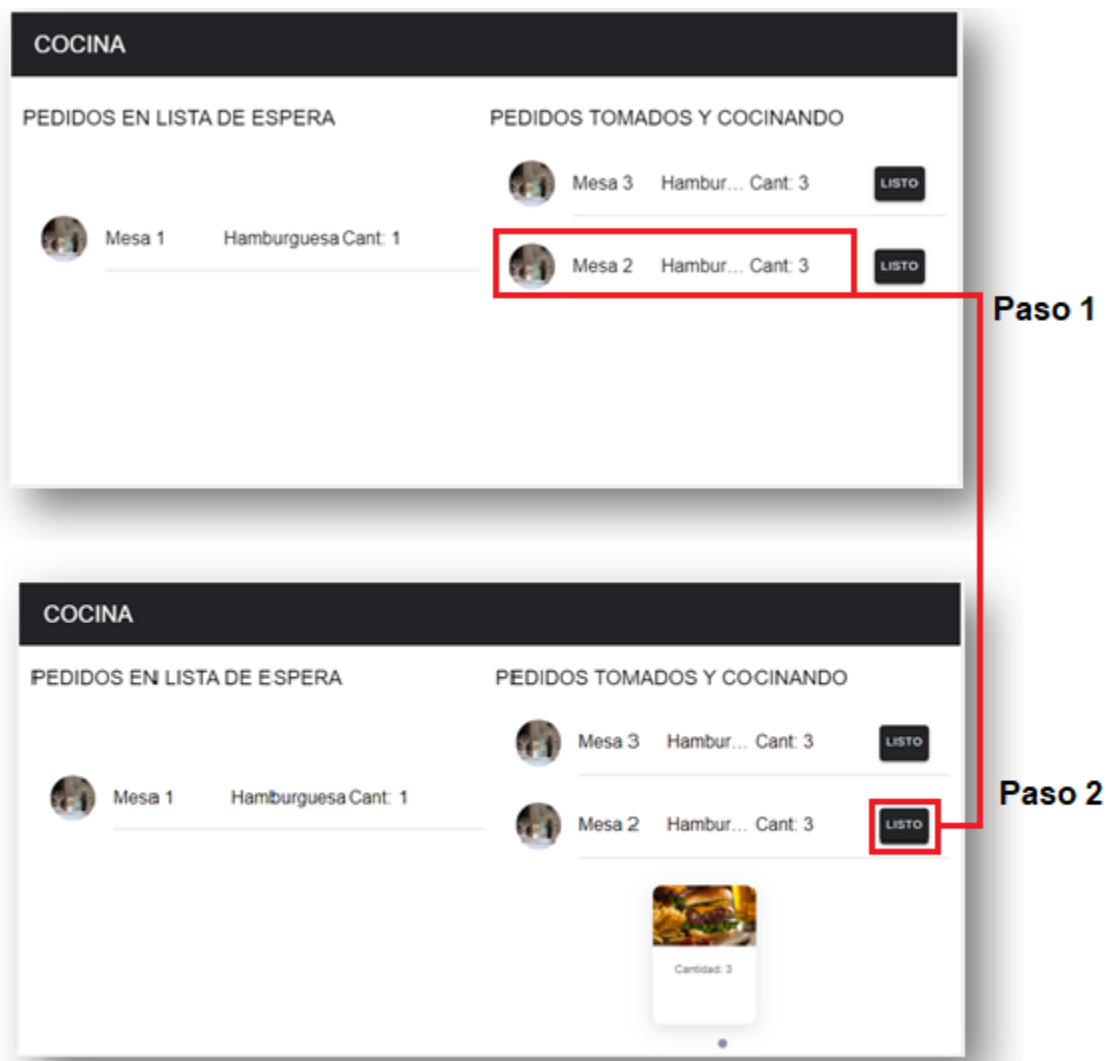


Figura 38. Secuencia pantallas del aplicativo de cocina para tomar el pedido.

Una vez se indique en el aplicativo de la cocina que todos los productos del pedido están listos y preparados para ser servidos, aparecerá en el aplicativo web un mensaje en la mesa de la respectiva orden. Este mensaje le permite saber al usuario que el pedido está listo y servido en la mesa respectiva, tal y como se observa en la figura 38.

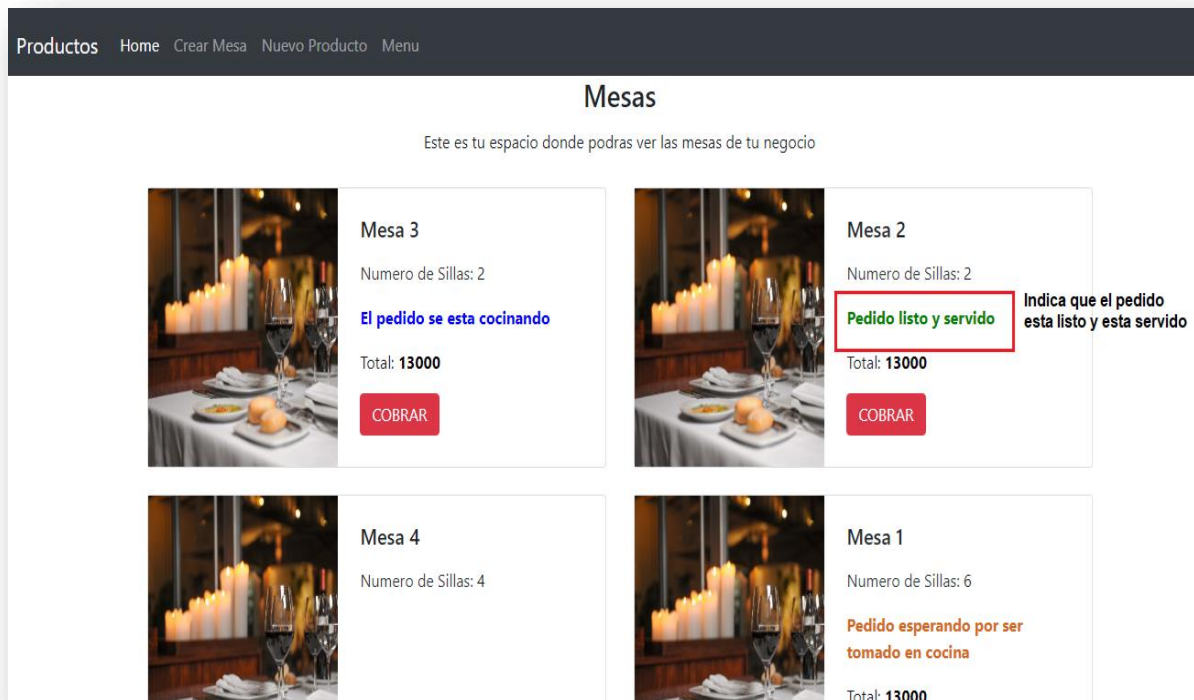


Figura 39. Aplicativo web con pedido listo en la mesa 2.

Después que el pedido esté listo, el siguiente paso será cobrar la mesa para que todo el ciclo culmine perfectamente. En la figura 40 se observan los pasos para cobrar una mesa en el aplicativo web. Para cobrar el costo total de una mesa, se debe dar clic en el botón COBRAR de la respectiva mesa, esto hará que aparezca una alerta, preguntando al usuario si está seguro que desea cobrar el pedido. Si se desea cobrar el pedido, se debe dar clic en el botón ACEPTAR de color azul, ubicado en el mensaje de la alerta, tal y como se observa en la figura 39.

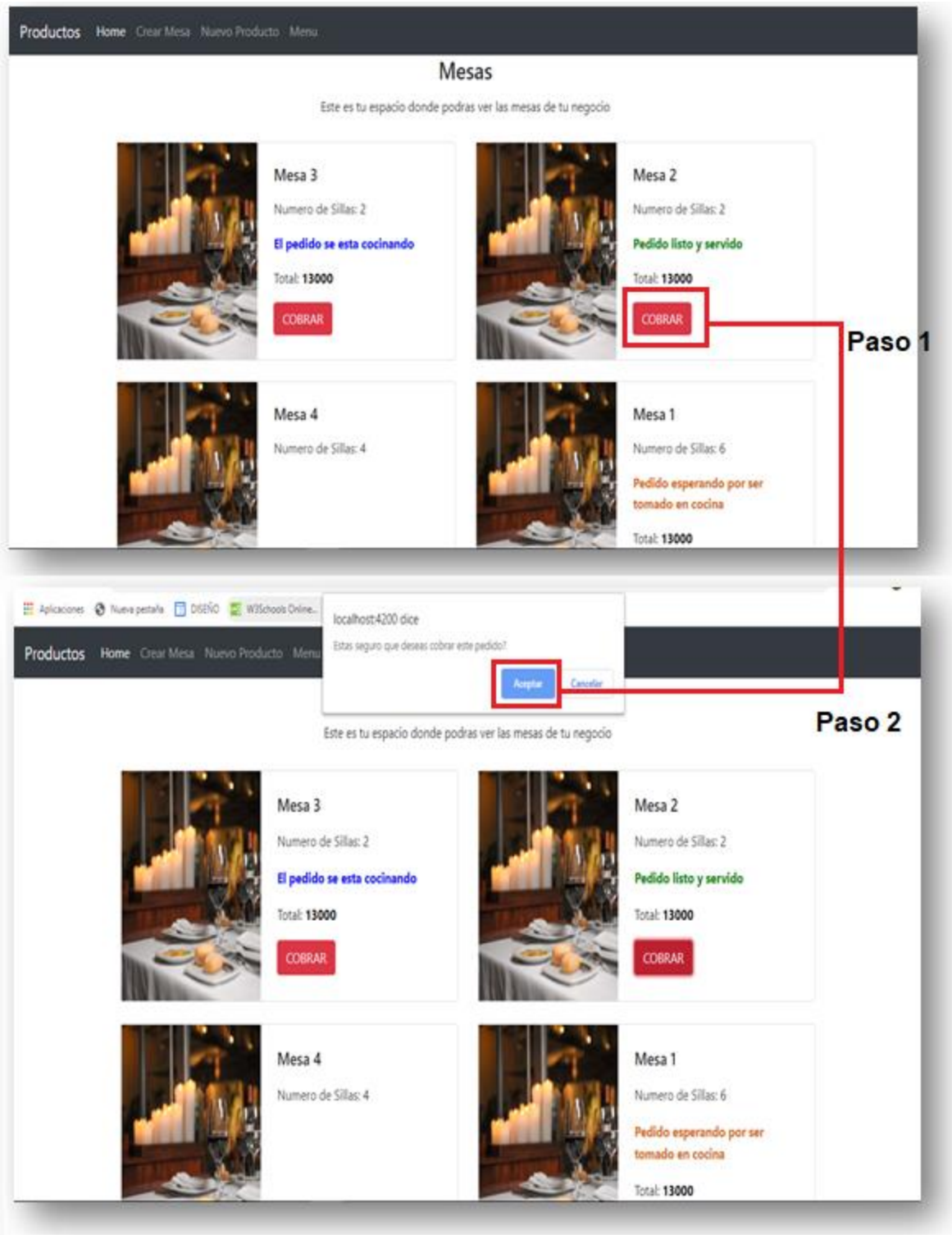


Figura 40. Secuencia pantallas del aplicativo de cocina para tomar el pedido.

Una vez cobrada la mesa en el aplicativo web, el respectivo pedido estará finalizado y la mesa volverá a estar vacía, en espera a que el usuario toda dicha mesa y realice una nueva orden. El aplicativo de la cocina muestra los pedidos que estén en espera o tomados y el aplicativo menú se encuentra en la pantalla principal, tal y como se observa en la figura 40.

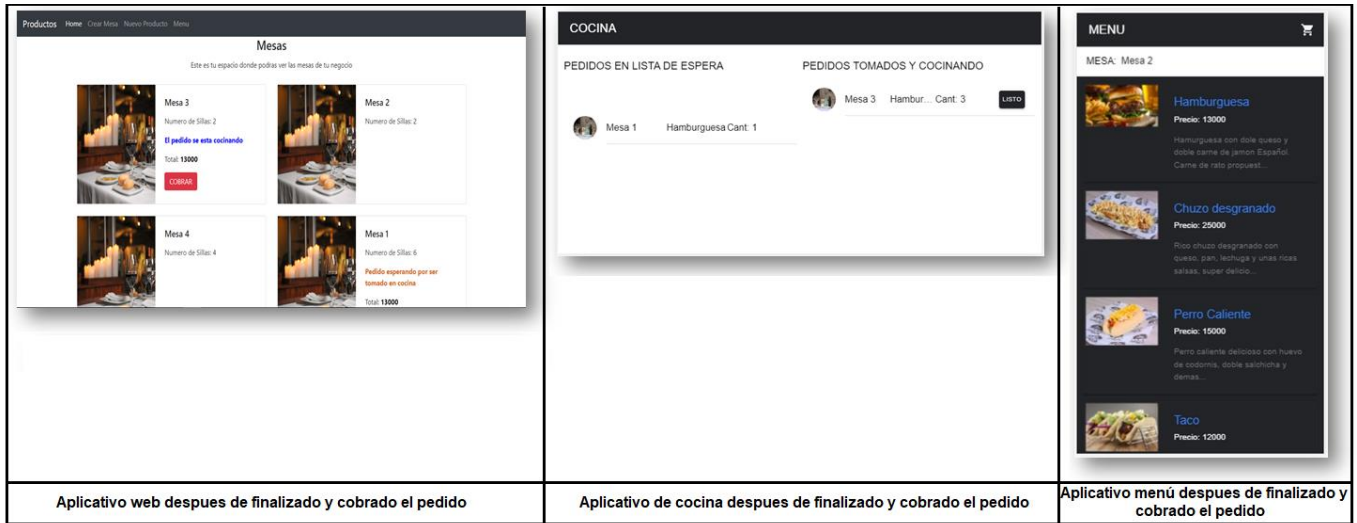


Figura 41. Secuencia pantallas del aplicativo de cocina para tomar el pedido.

ANÁLISIS Y RESULTADOS

Para obtener los datos requeridos de funcionalidad del aplicativo móvil de menú y aplicativo web administrativo, se realizaron 3 encuestas.

La primera encuesta se realizó a 40 personas entre 15 y 60 años de edad en las localidades de Pereira, Dosquebradas y Santa Rosa de Cabal, con el fin de comprobar la funcionalidad y usabilidad del aplicativo menú. Los resultados se pueden observar en la figura 41.

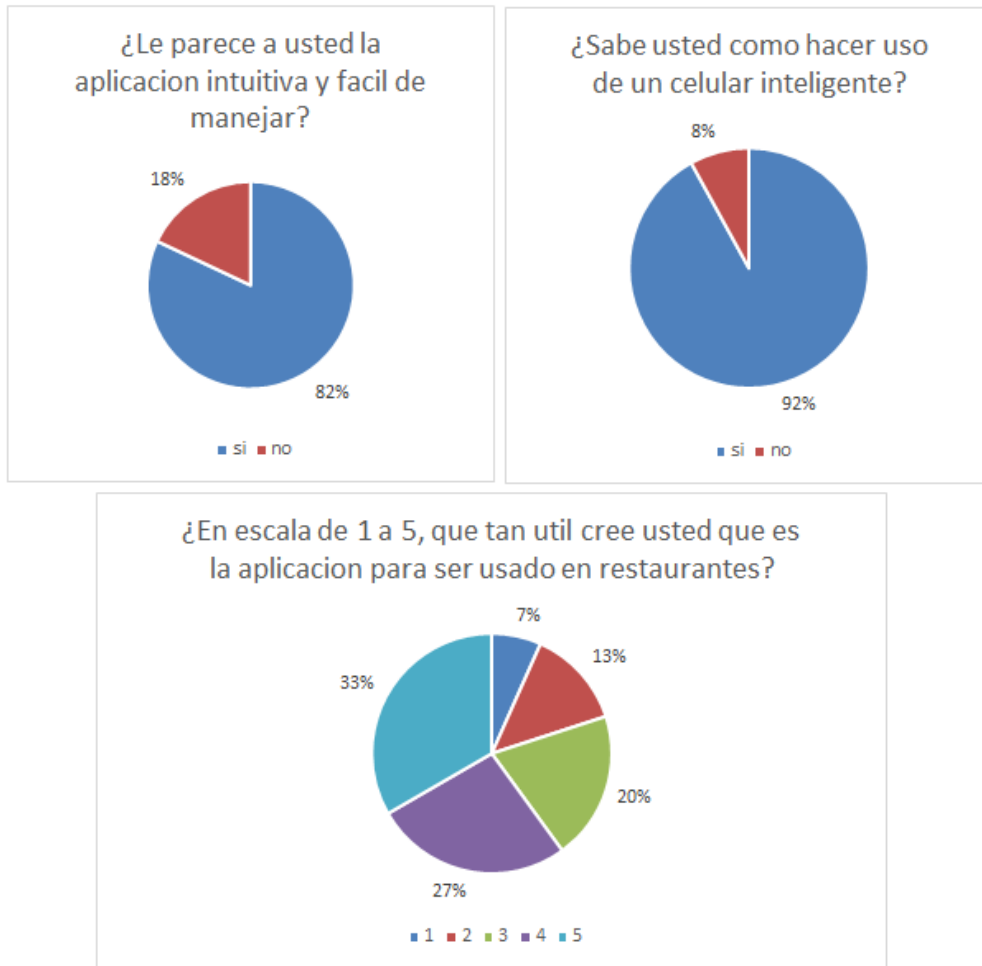


Figura 42. Resultados de encuestas para el aplicativo menú.

La segunda encuesta se realizó a 34 personas con estudios iguales o superiores a bachilleres entre 18 y 55 años de edad en las localidades de Pereira, Dosquebradas y Santa Rosa de Cabal, con el fin de comprobar la funcionalidad y usabilidad del aplicativo cocina. Los resultados se pueden observar en la figura 42.

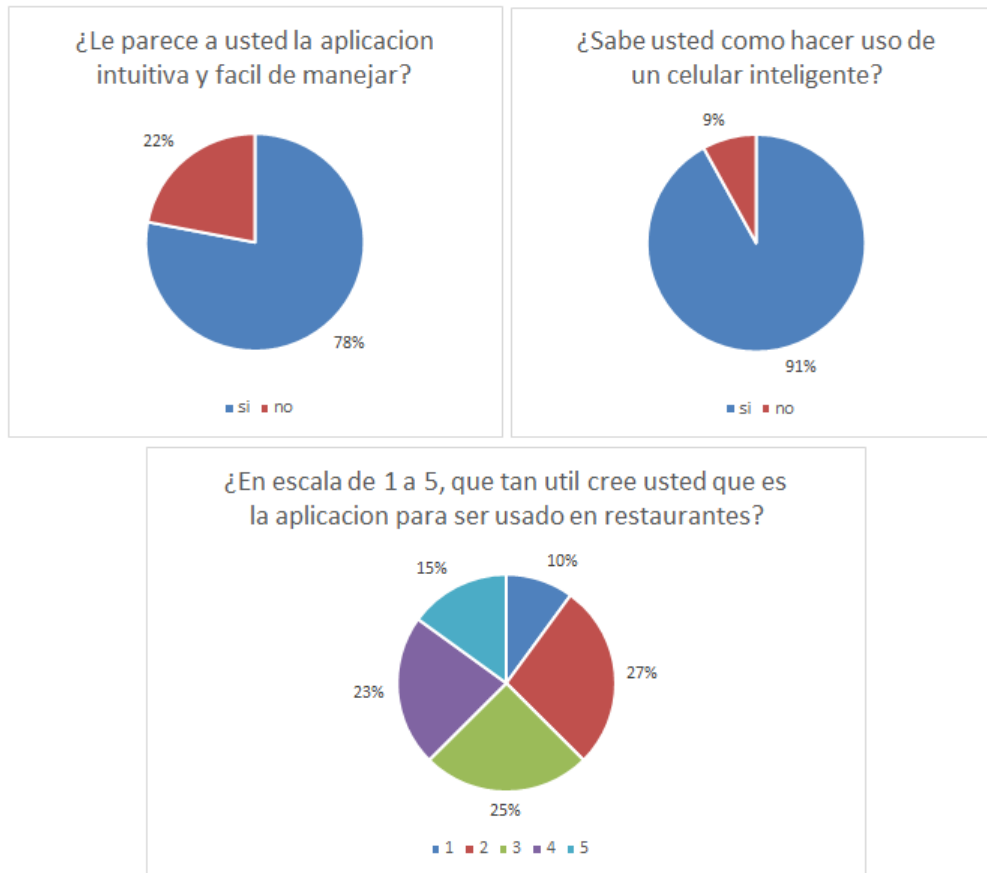


Figura 43. Resultados de encuestas para el aplicativo cocina.

La segunda encuesta se realizó a 32 personas con estudios superiores a bachilleres entre 18 y 50 años de edad en las localidades de Pereira, Dosquebradas y Santa Rosa de Cabal, con el fin de comprobar la funcionalidad y usabilidad del aplicativo web. Los resultados se pueden observar en la figura 43.

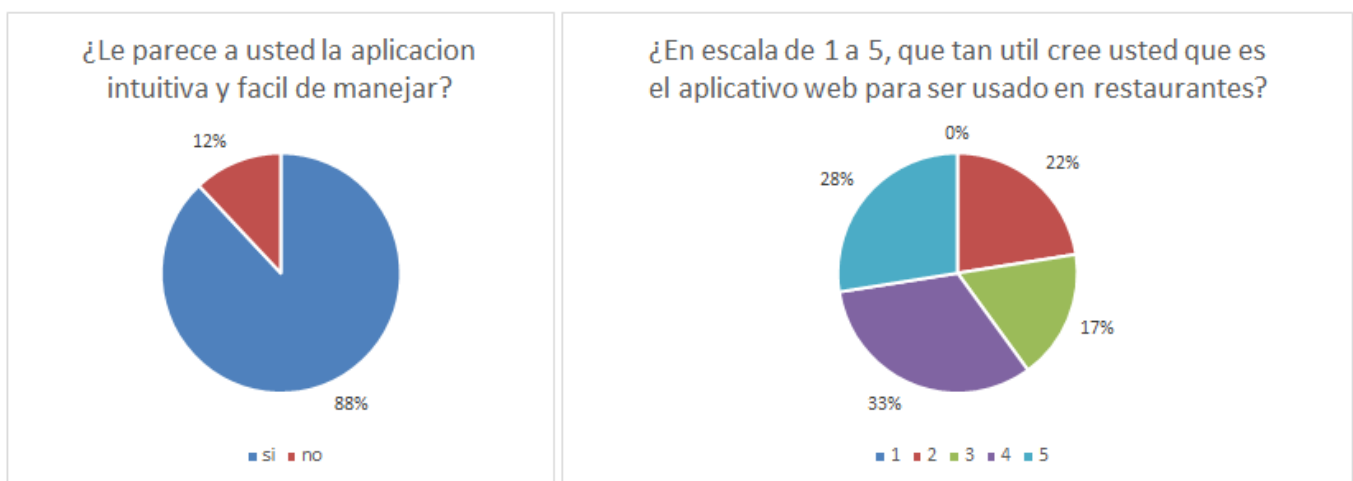


Figura 44. Resultados de encuestas para el aplicativo web.

A partir de las encuestas, se puede deducir que la mayoría de las personas entendieron y saben hacer uso del aplicativo menú, aplicativo cocina y aplicativo web. Se resalta que hubo una mayor cantidad de personas que no supieron utilizar el aplicativo cocina. Con respecto al aplicativo web, se tomó personas profesionales, ya que el uso de este aplicativo será exclusividad del administrador del restaurante, por esta razón se limitó la población a personas con estudios superiores a bachiller.

PRUEBAS CON DIFERENTES CELULARES

En las pruebas de trabajo del aplicativo móvil de menú y cocina se usaron diferentes móviles con marcas y versiones de sistemas operativos Android los cuales se muestran en la tabla 1, para lo cual se comprobó la compatibilidad con Android mayor a 6.0 sin mayores problemas, y se observó rapidez y una respuesta estable de las app's.

Para dispositivos IOS se instaló ionic devapp en un iphone 6 de versión IOS 12 para comprobar por medio de simulación la aplicación a través de Windows, aunque sin éxito, debido a los problemas de pluggins con ionic cordova en el sistema operativo Windows. Posteriormente, dados los inconvenientes se usó directamente una computadora MacOS mid 2012, por medio de la cual se pudieron instalar los aplicativos en el iphone 6 con sistema operativo IOS 12 sin problema. Se comprobó su funcionalidad en esta plataforma, lastimosamente, no se pudieron hacer pruebas en otras versiones.

Referencia Dispositivo	Versión Android
Xiomi redmi note 5	9.1
Huawei P8 lite	6.0
Xiaomi redmi note 7	9.0.0
Samsung Galaxy J5 prime	8.0.0
Moto G 5	8.1.0
Huawei p20 lite	9.0.0

Tabla 1. Móviles usados en las pruebas de funcionamiento.

CONCLUSIONES

- El sistema desarrollado cuenta con tres subsistemas que comunican las áreas de administración, cocina y servicio al cliente del restaurante, permitiendo así contar con una herramienta que permite al cliente tener claridad sobre lo que se va pedir, generando el pedido directamente en cocina, evitando errores y demoras en los pedidos. Además de poder realizar un monitoreo de los pedidos al administrador del negocio y así mejorar la productividad de su empresa. Ofreciendo de esta manera un valor agregado al dueño del negocio, ya que, los sistemas existentes conocidos solo se limitan al área de administración o al menú, más no comunican estas áreas entre sí y no tiene en cuenta a la cocina.
- Debido a la alta competencia de sistemas para restaurantes existentes, el sistema desarrollado ofrece una mejor flexibilidad, rapidez, mejor almacenamiento de datos y escalabilidad frente a los demás servicios TPV. El sistema desarrollado le ofrece al dueño del restaurante un sistema hecho a la medida, con la posibilidad de crecer dependientemente de los requerimientos del restaurante. Proveyendo de esta manera la posibilidad de mejor volumen de clientes, incluso, manejar varias cadenas de negocios, todo en un solo sistema. Por otro lado, el administrador y dueño del restaurante puede tener conocimiento del restaurante de una forma remota. Ofreciendo el sistema así, la facilidad de dirigir el negocio sin que el dueño esté presente en el lugar del restaurante.
- Al dividir el sistema en tres subsistemas pequeños se le quita carga computacional al hardware utilizado. Ya que, cada subsistema está implementado en dispositivos diferentes, toda la carga computacional se divide entre los diferentes dispositivos que el restaurante emplee. Por ejemplo, el aplicativo web es utilizado por el administrador en un computador, el aplicativo menú es utilizado por el mesero en un Smartphone y el aplicativo cocina es utilizado por el cocinero en una Tablet, de esta manera toda la carga computacional requerida es dividida entre un computador, un Smartphone y una Tablet. Por tanto, se le brinda mayor flexibilidad y rapidez al sistema.
- Por medio del aplicativo menú los meseros del restaurante pueden realizar una atención al cliente más eficiente. Puesto que el aplicativo menú cuenta con imágenes, precios y descripción de los productos ofrecidos, el mesero podrá prestar un servicio de calidad. De esta manera, el restaurante podrá mejorar sus ventas, ya que los meseros no solo se limitan a tomar las órdenes de los clientes, sino que también pueden vender los productos de una mejor manera.
- A la hora de realizar un desarrollo de software que cuenta con un app-móvil nativo y un app-web, es necesario seleccionar herramientas que tengan una estructura de programación parecidas, que permitan reutilización de código y que permitan un tiempo de desarrollo lo más corto posible. Es por esto que el uso de Angular y Ionic resulta muy útil para el desarrollo del aplicativo web y aplicativos móviles realizados en el presente proyecto. Puesto que, Ionic está construido en Angular, las similitudes de ambos Frameworks a la hora de programar resultan beneficiosas y permiten una excelente flexibilidad del sistema en cuanto a tiempo de desarrollo, debido a su filosofía de dividir los componentes de trabajo en su parte estética, estructural y comportamental.
- Puesto que Angular y Firebase son desarrollados por Google, la implementación de un servidor web por medio de Firebase es fácil y rápida. Por medio de los servicios de *Database* de Firebase se ahorró costos, tiempo de desarrollo y tiempo de aprendizaje, ya que hacer uso de servidores físicos propios no solo demanda mucho tiempo de desarrollo y aprendizaje, sino que exige realizar la compra o alquiler de servidores físicos.

- Por medio del trabajo actual, el administrador del restaurante tiene una herramienta que le permite tener un mejor conocimiento del estado de la cocina. Por medio de la filosofía de pedidos en espera, pedidos tomados y pedidos listos, el administrador que maneja el aplicativo web cuenta con una visión más amplia de la cocina y cada uno de los pedidos existentes. De tal modo, el administrador puede saber si existen pedidos que están siendo cocinados, si existen personas en determinada mesa que hayan realizado algún pedido, el tiempo entre órdenes, la eficiencia de la cocina respecto a cada preparación del menú, entre otros. Por otra parte, por medio del trabajo desarrollado, el administrador y dueño del restaurante puede tener conocimiento de su negocio de una forma remota por medio de internet. De esta manera, puede saber de manera cómoda y sencilla la situación actual del establecimiento. Ofreciendo el sistema así, la facilidad de dirigir el restaurante sin que el dueño esté presente en el lugar.
- En los resultados de las encuestas se tuvo un 22% de personas que no entendieron como usar el aplicativo de la cocina. Para dar solución a esto, se puede optar por mejorar a futuro dicho aplicativo, o dar una buena capacitación a los usuarios. Por otro lado, en la encuesta realizada para el aplicativo del menú se obtuvo un 18% de personas que no entendieron el uso del aplicativo. Por ende, para que todas las personas puedan hacer uso del aplicativo menú fácilmente, se puede agregar un corto tutorial o video, y se puede buscar a futuro que el aplicativo sea más amigable.

TRABAJOS FUTUROS

Una vez realizado este trabajo se proponen como trabajos futuros que pueden mejorar el sistema los siguientes desarrollos:

- Una herramienta de chat en la aplicación menú, con lo cual, el cliente podrá comunicarse con el administrador a través de un chat que se desplegará en los dispositivos, este chat será emergente y podrá estar en línea y en tiempo real. Con el fin de brindar un mejor servicio al cliente.
- Se dispondrá una opción de herramienta de pago en línea en la aplicación menú por medio de código QR, lo cual ofrecerá al cliente la facilidad de pagar desde la mesa por medio de su celular, sin tener que desplazarse hasta la caja principal. Además, de poder visualizar la cuenta y la factura detalladamente de los productos consumidos.
- Se tendrá una opción en la aplicación menú de solicitud de canciones desde la mesa, para que el cliente que haga el pedido pueda pedir canciones cada vez que hace un pedido.
- Desde el aplicativo web se tendrá la opción de enviar notificaciones a los clientes con el fin de promocionar sus productos y ofertas continuamente.
- Se integrarán servicios de analíticas personalizadas para determinar las tendencias de consumo de los clientes frente a los productos ofrecidos por el restaurante. De esta forma, se le ofrecerá la posibilidad a los dueños de restaurantes de mejorar su marketing, a través de datos reales de su negocio.

REFERENCIAS

- [1] Dinero, «Con flexibilización laboral sector gastronómico crearía 200.000 empleos,» Dinero, [En línea]. Available: <https://www.dinero.com/pais/articulo/como-esta-la-industria-gastronomica-en-colombia-este-2019/277044>.
- [2] P. M. Vargas, «Grupo Takami, el gusto por los restaurantes,» Portafolio, 19 11 2019. [En línea]. Available: <https://www.portafolio.co/negocios/empresas/la-estrategia-del-grupo-takami-534884>. [Último acceso: 30 12 2019].
- [3] Hiboutik, «Where to host mobile app backend?,» 2019. [En línea]. Available: <https://www.devteam.space/blog/where-to-host-mobile-app-backend>.
- [4] R. Tecnósfera, «Colombianos tocan su celular 2 mil veces al día en estas actividades,» El tiempo, 07 06 2019. [En línea]. Available: <https://www.eltiempo.com/tecnosfera/dispositivos/encuesta-de-consumo-movil-en-colombia-2019-389702>. [Último acceso: 30 12 2019].
- [5] Portafolio, «Colombia ya es el tercer mercado para Lenovo en Latinoamérica,» Portafolio, 10 12 2019. [En línea]. Available: <https://www.portafolio.co/negocios/empresas/colombia-el-segundo-mercado-para-lenovo-536394>. [Último acceso: 30 12 2019].
- [6] Foundation, «The Apache Software. About APACHE,» [En línea]. Available: http://httpd.apache.org/ABOUT_APACHE.html.
- [7] Devteam, «Where to host mobile app backend?,» [En línea]. Available: <https://www.devteam.space/blog/where-to-host-mobile-app-backend>. [Último acceso: 2019].
- [8] Monteiro, Hands-on full stack web development with angular 6 and laravel 5, 2a edition. packt publishing ltd., livery place, 35 livery street birmingham b3, uk, 2018.
- [9] Freeman, Pro Angular, United Kingdom, Milton Keynes: 2a edtion. apress, 2017.
- [10] N. Murray, ng-book The complete book on angular 8, San Francisco: Fullstack.io, 2018.
- [11] J. Moreno, «El nacimiento de los Web Services,» 2016. [En línea]. Available: URL <https://programar.cloud/post/el-nacimiento-de-los-web-services/>.
- [12] Bbva, «Breve historia de las apis: del comercio electrónico a la era móvil,» Bbva, 2018. [En línea]. Available: <https://bbvaopen4u.com/es/actualidad/breve-historia-de-las-apis-del-comercio-electronico-la-era-movil>.
- [13] Serinfo, «Historia del tpv ¿la conoces?,» [En línea]. Available: <https://www.serinfo.net/historia-del-tpv-la-conoces/>.
- [14] M. Libre, «Historia de la web: Normas y recomendaciones,» Mc Libre, 2019. [En línea]. Available: <http://www.mclibre.org/consultar/htmlcss/otros/historia.html>.
- [15] K. S, Full stack angularjs for java developer, Maha- rashtra, India: 1a edition. apress, bangalore, 2017.
- [16] Cheng, Build mobile apps with ionic 2 and firebase, segunda ed., Aucklan: apress.
- [17] J. Jude, Learn ionic 2, Karnataka, india: 1a edition. Apress, Bangalore, 2017.
- [18] I. Lake, Professional Android, Indianapolis, Usa: 4a edition. John Wiley Sons Inc, Crosspoint Boulevard , 2018.
- [19] Oracle, Openair Soap Api Reference Guide, Arizona, Usa: 1a edition. Oracle Publishing, Pinal County, 2019.
- [20] TECHCRONUS, «Benefits Of Ionic Framework App Development,» TECHCRONUS, 12 11 2018. [En línea]. Available: <https://www.techcronus.com/blog/benefits-ionic-framework-app-development/>. [Último acceso: 02 06 2019].
- [21] B. Behlendorf, «Collabnet,» [En línea]. Available: <https://web.archive.org/web/>

20051210070436/http://www.collab.net/about/estaff/brian.html.

- [22] B. Marc, «el empresario igualitario detrás de salesforce,» Amexcorporate, [En línea]. Available: <https://www.amexcorporate.com.ar/multitaskers/nota.php?id=696&cat=8>.
- [23] M. Barrera, «guía Metodológica para el proceso de validación y verificación de requerimientos para el usuario final,» Bogota, 2015.
- [24] J. Wilken, Ionic in action, Manning Publications, 2015.
- [25] L. Moroney, The definitive guide to firebase, Seattle, USA: Apress, 2017.
- [26] A. Calderon, «Tiendas virtuales y comercio web,» [En línea]. Available: <https://www.tiendasvirtualesycomercioweb.com/index.php>. [Último acceso: 02 07 2019].
- [27] P. Kyllönen, «Mobile-D: an agile approach for mobile application development,» ACM DIGITAL LIBRARY, [En línea]. Available: <https://dl.acm.org/citation.cfm?id=1028736>. [Último acceso: 02 06 2019].
- [28] G.B, «¿Qué es Apache? Descripción completa del servidor web Apache,» Tutorial Hostinger, 2019. [En línea]. Available: <https://www.hostinger.co/tutoriales/que-es-apache/>. [Último acceso: 01 06 2019].
- [29] G. Segato, «Una introducción a Volley,» evantotuts, 13 05 2015. [En línea]. Available: <https://code.tutsplus.com/es/tutorials/an-introduction-to-volley--cms-23800>. [Último acceso: 06 06 2019].
- [30] Enesimus, «trabajando con JSON,» MDN web docs, 11 08 2019. [En línea]. Available: <https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON>. [Último acceso: 18 11 2019].
- [31] Wallace, Android apps for absolute beginners, California: 1a edition. apress, lompoc, 2017.
- [32] C. Pinzón, Meta Análisis de los estilos de arquitectura de software orientados a la web, Bogota, Colombia: Universidad Católica de Colombia, 2017.
- [33] R. Varshney, «Entrepreneur,» [En línea]. Available: <https://www.entrepreneur.com/article/266137>. [Último acceso: 02 06 2019].